ANO I Nº 07
NCZ$ 1,50

CPU

O CURSOR NO MSX   PAY LOAD   SINTEVOZ

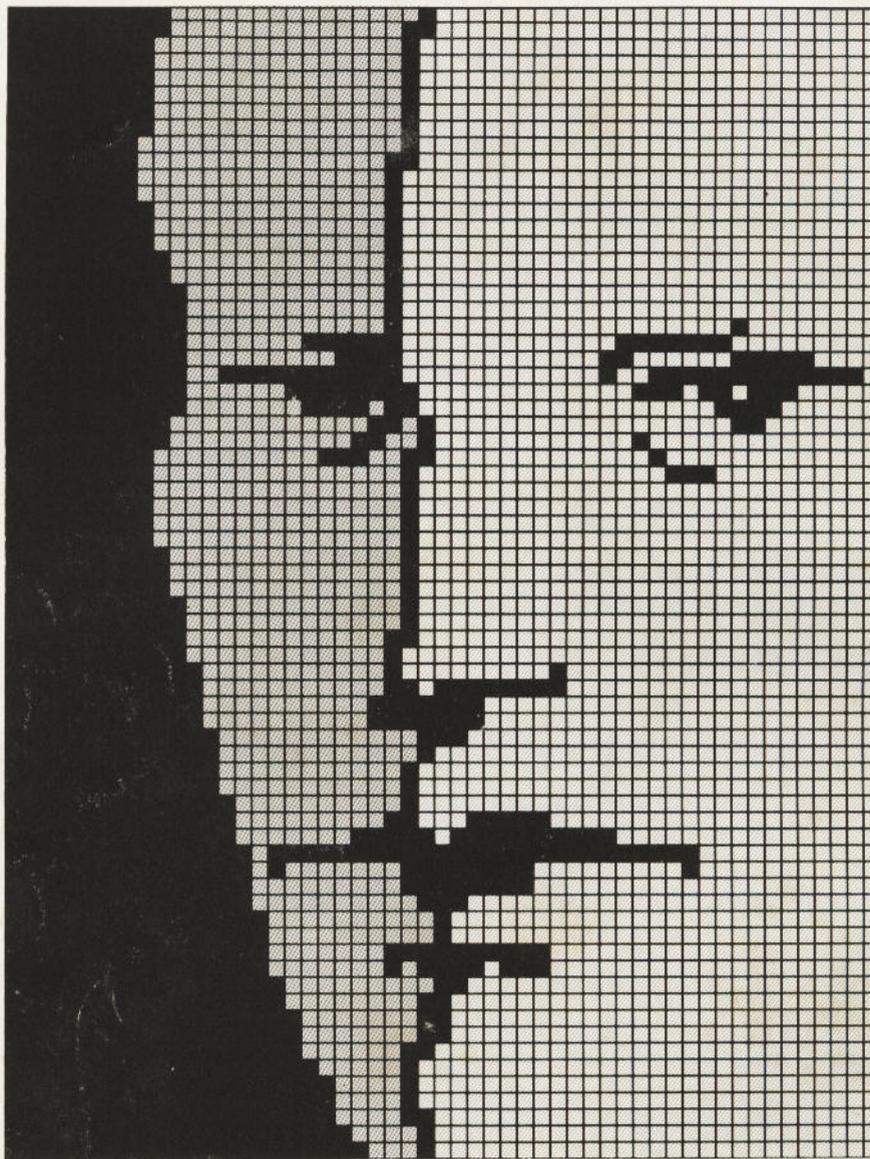
BUS EXPANSION E RESET NO EXPERT

3ª Fenasoft

OS MAIORES AVANÇOS DA INFORMÁTICA BRASILEIRA FORAM CONSEGUIDOS NESTA FEIRA.

Em março de 89, você vai saber porquê a informática tem conquistado grandes mercados no Brasil.

De 14 a 17, no Riocentro, Rio de Janeiro, a 3ª Fenasoft apresenta todos os lançamentos e produtos das áreas de software e hardware de mais de 200 empresas, na maior feira de informática do Brasil.

É a grande oportunidade para empresários divulgarem e comercializarem programas e equipamentos, num evento que tem competência comprovada.

Só nas ultimas duas edições, a Fenasoft foi responsável por mais de 400 milhões de dólares em negócios efetivamente realizados.

Uma prova de que nenhuma empresa resiste a um bom programa de informática. Pois na era da competência, adquirir software e hardware é um grande passo para que projetos empresariais virem dinheiro.

U$ 400 MILHÕES EM NEGÓCIOS REALIZADOS.

Patrocínio:
ASSESPRO/NACIONAL
SEI
SECRETARIA MUNICIPAL DE DESENVOLVIMENTO ECONÔMICO DO RIO DE JANEIRO
ABICOMP
Apoio:
EMBRATEL Empresa do SISTEMA TELEBRÁS

EMPRESA AÉREA OFICIAL

FICHA DE SOLICITAÇÃO

DESEJO RECEBER

- [ ] VISITA REPRESENTANTE
- [ ] MATERIAL INFORMATIVO
- [ ] 3ª FEIRA INTERNACIONAL DO SOFTWARE
- [ ] 3º CONGRESSO INTERNACIONAL DA TECNOLOGIA DO SOFTWARE, TELEMÁTICA E INFORMAÇÃO

NOME: ______

END: ______ FONE: ______

EMPRESA: ______ CEP: ______

Promoção:
FAIR- FEIRAS E EMPREENDIMENTOS LTDA.

RIO DE JANEIRO
Rua Olegário Maciel nº 175, cb. 01, CEP-22621, Tels. (021) 399-9077/5185, Tlx. (21) 38446, Rio de Janeiro, RJ.

SÃO PAULO
Rua Hungria nº 664, 10º andar, cj. 101, CEP-01455, Tel. (011) 815-2315, Tlx. (11) 37328, São Paulo, SP.

SANTA CATARINA
Rua Ivo Reis Montenegro nº 221, lj. 04, CEP-88010, Pq. Residencial Floresta, Tel. (0482) 46-3663, Tlx. (481) 213, São José, SC.

RIOCENTRO - RIO DE JANEIRO: 14 A 17 DE MARÇO DE 1989

EVENTO OFICIALIZADO PELO M.I.C. - C.D.C.

ÁGUIA INFORMÁTICA LTDA
AV. N. SRA. DE COPACABANA 605/804
COPACABANA
22040 - RIO DE JANEIRO - RJ
TELEFONE: 021-235.3541

DIRETOR RESPONSÁVEL
GONÇALO R. F. MURTEIRA

DIRETOR COMERCIAL
JOSÉ GERALDO J. NETO

JORNALISTA RESPONSÁVEL
DOLAR TANUS
REGISTRO 430-RS

COLABORADORES
PEDRO HENRIQUE GAMA
PAULO MARQUES FIGUEIRA
SÉRGIO GUY PINHEIRO ELIAS
PAULO ROBERTO PINHEIRO ELIAS
BRUNO MARRUT
JÚLIO VELLOSO
SÉRGIO DURIC CALHEIROS

REVISÃO DE TEXTO
LAURA MARIA PINTO

CAPA
JOSÉ AGUILERA

PROJETO GRÁFICO
LUCIANA MONTENEGRO

ADMINISTRAÇÃO
JOSÉ A. NASCIMENTO

ASSINATURAS
MARILZA LOURENÇO

FOTOLITOS
ESCALA FOTOLITOS

IMPRESSÃO
EDITORA SANTUÁRIO

DISTRIBUIÇÃO
FERNANDO CHINAGLIA DISTRIBUIDORA

CPU é uma publicação da Águia Informática. Todos os direitos reservados. Proíbida a reprodução parcial ou total do conteúdo desta revista por qualquer meio sem autorização expressa da editora.
Os artigos assinados são de total e única responsabilidade dos autores.
Os circuitos, dispositivos, componentes, etc., descritos na revista podem estar sob a proteçaõ de patentes. Os circuitos publicados só poderão ser confeccionados sem qualquer fim lucrativo.
Os programas apresentados aos leitores, mesmo se fornecidos em disquete, são de propriedade dos autores, cabendo a eles todos os direitos previstos em lei.

ÍNDICE

SEÇÕES

A GERAÇÃO DO CURSOR NO MSX

RENATO DA SILVA OLIVEIRA

Alguns micros apresentam na tela um cursor piscante, cuja frequência do acender e apagar pode ser regulada através da alteração do conteúdo de alguma variável do sistema. Nos micros MSX o cursor é gerado de modo fixo, mas pode ser alterado através de software para piscar.

Normalmente, o processo pelo qual o cursor é gerado nos MSX envolve basicamente seis variáveis do sistema:

CSRX	em &HF3DD	coordenada horizontal do cursor.
CSRY	em &HF3DC	coordenada vertical do cursor.
CURSAV	em &HFBCC	rascunho para armazenar o caractere sob o cursor.
LINWRK	em &HFC18	rascunho de 40 bytes.
CSRSW	em &HFCA9	indicador de cursor "ligado" (0) ou "desligado" (255).
CSTYLE	em &HFCAA	indicador de cursor cheio ou pela metade.

Figura 1 - CARACTERE "A" NA TELA

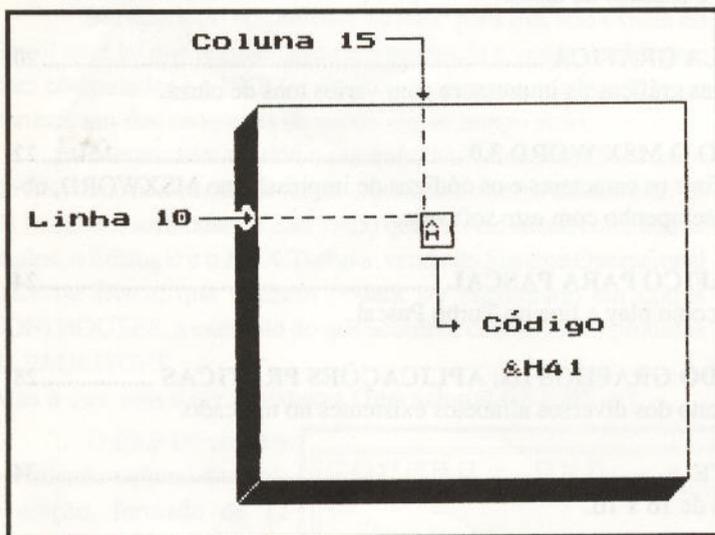



Figura 2 - CONTEÚDO DE CURSAV

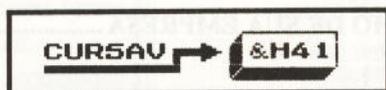
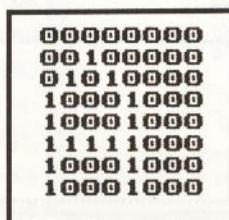



Figura 3 - DESENHO DO "A" NA VRAM

```
00000000
00100000
01010000
10001000
10001000
11111000
10001000
10001000
```

O cursor é gerado apenas quando a variável CSRSW contém zero.

Para gerar o cursor, o caractere apontado na tela pelas variáveis CSRX e CSRY tem seu código copiado da VRAM para a variável CURSAV (lembre-se de que quando a tela está "limpa", na verdade, ela está repleta de caracteres nulos, de código 0).

A seguir, seu desenho (8 bytes) é lido da VRAM e copiado para a variável LINWRK. Se CSTYLE for 0, os 8 bytes são invertidos, se não, apenas os 4 bytes inferiores o são.

Finalmente, os 8 bytes alterados em LINWRK são copiados para a posição do cursor (código 255) na tabela de caracteres da VRAM e código 255 é gravado na tela, exatamente na posição apontada por CSRX e CSRY.

Apenas a título de ilustração, vamos supor que o caractere sobre o qual o cursor esté posicionado é a letra "A", na linha 10 e coluna 15 da SCREEN 0. Se CSRSW=0 e CSTYLE=0, o processo de geração do cursor é ilustrado pela seguinte sequência (figuras 1 a 5).

Figura 4 - DESENHO APÓS CÓPIA E INVERSÃO EM LINWRK E DESENHO DO CURSOR NA VRAM

```
LINWRK ⟶ 11111111
         11011111
         10101111
         01110111
         01110111
         00000111
         01110111
         01110111
```

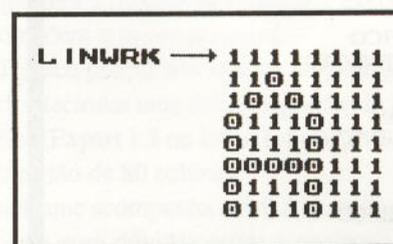
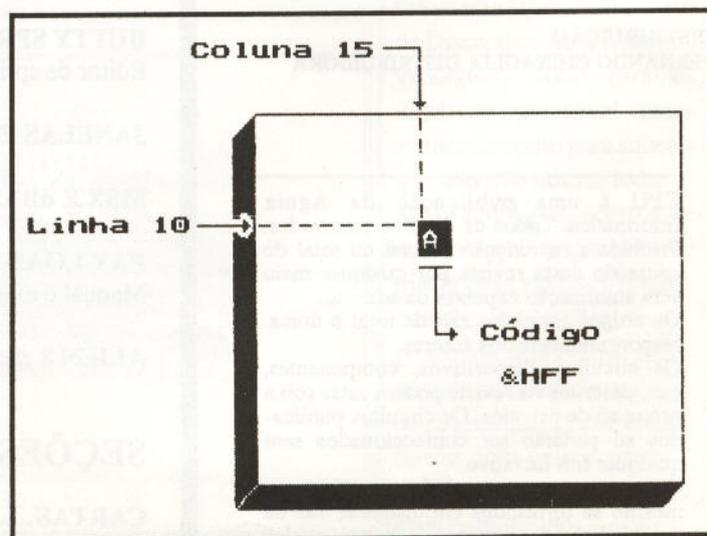



Figura 5 - CURSOR NA TELA

A listagem da figura 6 foi digitada usando-se o editor de textos MSX WRITE. Você poderá usar qualquer outro editor que grave os dados em ASCII, como o SCED (ou MED, na versão mais recente), o WordStar, o Deletra, etc. Cuidado se você for usar o Tass Word original, pois seu formato de gravação é peculiar. Existem, entretanto, versões adaptadas que também gravam em ASCII. Lembre-se ainda que os comentários não precisam ser digitados!

Uma vez digitado o texto do programa, teremos que compilá-lo. Ele está escrito de modo a ser compilado diretamente pelo M80 da Microsoft. Entretanto, se você quiser, poderá adaptá-lo facilmente para uso com o Coral Simple ASM (Cocar e Mega Assembler, nas "versões" nacionais). Para tanto, basta numerar as linhas do programa, substituir a pseudo-instrução ".Z80" por "ORG 0D000H" e acrescentar os dois pontos (":") antes das pseudo-instruções "EQU".

Para fazer a compilação com o M80, grave o programa com o nome "CURSOR.MAC" e comande, a partir do MSXDOS:

A>M80 =CURSOR

A seguir, se tudo correu bem, use o L80 com a seguinte sintaxe:

A>L80 /p:d000,cursor,cursor/n/x/e

Finalmente, use o comando BSAVE do MSXDOS TOLL's:

A>BSAVE cursor.hex cursor.bin

Agora, experimente comandar:

A>dir cursor.*

Você deverá ter em disco quatro arquivos:

CURSOR.MAC
CURSOR.REL
CURSOR.HEX
CURSOR.SIN

Se quiser, apague os arquivos terminados em ".REL" e em ".HEX", preservando os outros dois.

O arquivo "CURSOR.BIN" é o programa em Linguagem de Máquina, para ser carregado e executado através do BASIC.

Para fazer a compilação com o Coral Simple ASM, basta comandar:

>AOU

Para salvá-lo em disco, vá para o BASIC com o comando ">BA" e comande:

BSAVE"CURSOR.BIN",&HD000,&HD080

Agora, com o programa em L.M. já carregado na memória, execute-o com os comandos:

DEFUSR=&HD000:POKE0,USR(5)

Observe que, se tudo correu bem até aqui, o cursor deverá estar piscando. Experimente alterar o valor do parâmetro passado na função USR. Ao invés de 5, use outros valores. Esse parâmetro regula a frequênci do "acender" e "apagar" do cursor. Se você passar o parâmetro 0, o cursor voltará ao estado normal, sem piscar.

Experimente também levar o cursor piscando para a última linha da tela com as teclas de setas. Não se asuste se ele parar de piscar, pois é isso o que deve ocorre sempre que ele estiver na última linha. Teste também os comandos "KEY ON" e "KEY OFF".

Agora que já temos o progrma digitado e funcionando, vamos tentar entender a listagem em assembly. Antes de mais nada, tente acompanhar os comentários na própria listagem.

A primeira parte do programa recebe o parâmetro passado pelo BASIC, testa se ele é um inteiro e o insere na variável VELOCI. A seguir, se o parâmetro for zero, o hook HKEYI é restabelecido com um RET (0C 9H), desligando assim a rotina para piscar. Se o parâmetro for um inteiro diferente de 0 e entre 1 e 255, o hook HKEYI é apontado para a rotina INÍCIO.

A rotina INICIO será executada sempre que uma interrupção ocorrer. Inicialmente todos os registros do Z80 são salvos e um teste é realizado para saber se o cursor está na última linha útil da tela. Se estiver, a "piscadá" do cursor é desligado. Isso é feito para evitar complicações com a rotina de SCROLL da ROM. Sempre que um SCROLL for executado, se o cursor estiver piscando, poderão ocorrer efeitos estranhos no vídeo. Pode-se evitar isso de várias formas mas, para não tornarmos o programa muito grande, optamos por simplesmente desligar a "piscada" quando o SCROLL ocorre. Note que, quando as teclas de funções são mostradas, a última linha da tela é a vigésima terceira e a variável CNSDFG contém o valor &HFF.

O trecho seguinte do programa verifica se está no momento de inverter o cursor ou não. Isso é feito comparando o valor de CONTAD com o valor de VELOCI. Se foremm iguais, a inversão do cursor é realizada e o valor de CONTAD é zerado. Caso contrário, a variável CONTAD é incrementada e a execução passa para a rotina de retorna.

A rotina de inversão é muito simples, invertendo, apenas cada um dos bytes do desenho do cursor (código 255) na tabela de caracteres da VRAM. Note que o valor 2040 é a posição relativa do primeiro byte do cursor dentro da tabela e a variável TXTCGP aponta para o início da tabela da VRAM. A rotina LACO é repetida 8 vezes e é bastante simples. Quando ela é executada, B contém o valor 8 e HL áponta para o endereço da VRAM com o primeiro byte do desenho do cursor. As rotinas RDVRM e WRTVRM, apesar de existirem no BIOS, foram refeitas e funcionam de modo análogo: o par HL deve apontar o endereço da VRAM e o byte lido ou gravado deve estar no registro A.

Sendo assim, o LACO apenas lê cada um dos 8 bytes do desenho do cursor, os inverte e os grava novamente na mesma posição.

Aos amigos ex-usuários do TRS-80, lembramos mais uma vez para que atentem à versatilidade do MSX!

Se você está interessado no efeito do cursor piscando mas não quiz se arriscar `as "complicações" com a compilação, "linkagem" e binarização, basta digitar e rodar o programa em BASIC listado na figura 7. Ele gera a rotina em linguagem de máquina e a executa, passando o parâmetro 5 na função USR.

FIGURA 7 - PROGRAMA EM BASIC

```
1000 SCREEN 0 : WIDTH 40 : KEY OFF
1010 FOR F=&HD000 TO &HD080
1020   READ A$ : POKE F,VAL("&H"+A$)
1030 NEXT F : CLEAR 200,&HD000
1040 DEFUSR0=&HD000 : POKE 0,USR0(5)
1050 '
1200 DATA F3,FE,02,C0,23,23,7E,32
1210 DATA 7F,D0,FE,00,20,07,3E,C9
1220 DATA 32,9A,FD,18,0B,21,22,D0
1230 DATA 22,9B,FD,3E,C3,32,9A,FD
1240 DATA FB,C9,F3,D9,3A,DE,F3,E6
1250 DATA 01,47,3A,DC,F3,80,FE,18
1260 DATA 28,27,21,7F,D0,3A,7E,D0
1270 DATA BE,28,06,3C,32,7E,D0,18
1280 DATA 18,2A,B7,F3,01,F8,07,09
1290 DATA 06,08,CD,5B,D0,2F,CD,69
1300 DATA D0,23,10,F6,3E,00,32,7E
1310 DATA D0,D9,C9,7D,D3,99,7C,E6
1320 DATA 3F,D3,99,CD,7B,D0,DB,98
1330 DATA C9,F5,7D,D3,99,7C,E6,3F
1340 DATA F6,40,D3,99,CD,7B,D0,F1
1350 DATA D3,98,C9,E3,E3,C9,00,05
1360 DATA 00,58,53,57,20,52,53,4F
```

Renato da Silva Oliveira é diretor da XSW Publicações e Planejamento e Sistemas Ltda e autor de vários livros sobre a linha MSX.

Note que quando as variáveis CSRX ou CSRY forem alteradas, forçando a mudança de posição do cursor, o caractere armazenado em CURSAV é repassado para a sua posição original na tela, restabelecendo assim o seu aspecto antes da formação do cursor.

Como pudemos notar, não existe um desenho "fixo" para o caractere cursor nos micros MSX. Seu desenho é sempre redefinido para apresentar o inverso do caractere sobre o qual ele se encontra.

Agora vamos ilustrar um processo pelo qual podemos interferir na formação do cursor.Para isso teremos que recorrer ao uso de um hook da rotina de interrupção do micro. O primeiro hook chamado pela rotina de interrupção é o HKEYI, em &HFD9A. Vamos apontá-lo para um pequeno programa em L.M. que vai alterar a formação do cursor.

Observe a listagem da figura 6. Ela apresenta a nossa rotina com alguns comentários sobre seu funcionamento.

Convém ter em mente que ela é bastante específica, funcionando apenas na SCREEN 1 e ainda com uma restrição para a última linha da tela.

Existem outras formas mais rápidas, mais curtas e mais genéricas para se produzir o mesmo efeito, usando algumas rotinas que já existem prontas na ROM do micro. Preferimos, entretanto, dar mais atenção ao aspecto ilustrativo que ao de eficiência e optamos pela rotina ora apresentada.

Vamos ver inicialmente como deve ser gerado o programa em L.M. a partir da listagem em assembly da figura 6.

Figura 6 - LISTAGEM EM ASSEMBLY

```
;--------------------------------
; PROGRAMA PARA CURSOR "PISCANTE"
; FAZENDO A INVERSAO  (R.S.O.-89)
; ))) Somente para a SCREEN 0 (((
;--------------------------------
;
        .z80            ;compilacao em assembly Z80
;
;--------------------------------
; Variaveis do sistema e hooks
;--------------------------------
;
TXTCGP  EQU     0F3B7H          ;inicio da tabela de caracteres na VRAM
CSRY    EQU     0F3DCH          ;posicao vertical do cursor
CNSDFG  EQU     0F3DEH          ;indicador de mostra (0) ou nao mostra (255) teclas de funcoes
HKEYI   EQU     0FD9AH          ;primeira hook de interrupcao
;
;--------------------------------
; Rotina para ligar ou desligar
;--------------------------------
;
        DI                      ;desabilita a interrupcao
        CP      2               ;verifica se parametro e inteiro
        RET     NZ              ;retorna se nao for
;
        INC     HL              ;aponta HL para o parametro passado
        INC     HL
        LD      A,(HL)          ;carrega parametro no acumulador
        LD      (VELOCI),A      ;carrega parametro como VELOCIdade
        CP      0               ;verifica se velocidade e zero e
        JR      NZ,SETHOK       ;se nao for, liga a rotina com a hook
        LD      A,0C9H          ;mas se for, desliga a rotina na hook
        LD      (HKEYI),A
        JR      SETH01
;
SETHOK: LD      HL,INICIO       ;aponta hook para a rotina
        LD      (HKEYI+1),HL
        LD      A,0C3H
        LD      (HKEYI),A
SETH01: EI                      ;habilita a interrupcao
        RET                     ;retorna para o BASIC
;
;--------------------------------
; Rotina da interrupcao
;--------------------------------
;
INICIO: DI                      ;desabilita a interrupcao
        EXX                     ;salva todos os registradores
        LD      A,(CNSDFG)      ;verifica se as teclas de funcoes estao na tela e
        AND     1               ;se estiverem, carrega o acumulador com 1, se nao, com 0
        LD      B,A             ;carrega B com o acumulador
        LD      A,(CSRY)        ;carrega a coordenada vertical do cursor no acumulador
        ADD     A,B             ;soma coordenada vertical com 0 (teclas ausentes) ou 1
                                 (teclas presentes)
        CP      24              ;verifica se o cursor esta na ultima linha da tela
        JR      Z,RETORN        ;e se estiver, retorna

        LD      HL,VELOCI       ;aponta HL para a VELOCIdade
        LD      A,(CONTAD)      ;carrega contador no acumulador
        CP      (HL)            ;compara velocidade com contador e
        JR      Z,INVERT        ;se iguais, inverte o cursor
        INC     A               ;mas se nao, incrementa o contador
        LD      (CONTAD),A      ;e
        JR      RETORN          ;retorna
;
INVERT: LD      HL,(TXTCGP)     ;aponta HL para inicio da tabela de caracteres da VRAM
        LD      BC,2040         ;carrega BC com 255*8 (inicio do caractere cursor na tabela)
        ADD     HL,BC           ;aponta HL para inicio do caractere cursor na VRAM
        LD      B,8             ;carrega B (contador do LACO) com 8
LACO:   CALL    RDVRM           ;le um byte do desenho do cursor
        CPL                     ;inverte o byte lido
        CALL    WRTVRM          ;grava o byte invertido
        INC     HL              ;incrementa HL
        DJNZ    LACO            ;decrementa B e repete o LACO se ele nao for zero
;
        LD      A,0             ;carrega 0 no acumulador
        LD      (CONTAD),A      ;zera o contador
;
RETORN: EXX                     ;restabelece os registradores
        RET                     ;retorna
;
;--------------------------------
; Le o endereco apontador por HL
; na VRAM e o devolve em A
;--------------------------------
;
RDVRM:  LD      A,L             ;prepara VDP para ler o
        OUT     (099H),A        ;endereco apontado em
        LD      A,H             ;HL na VRAM
        AND     03FH
        OUT     (099H),A
        CALL    TEMPO           ;espera tempo para o VDP
        IN      A,(098H)        ;le o byte da VRAM
        RET                     ;retorna
;
;--------------------------------
; Grava o byte do acumulador no
; endereco da VRAM apontado por HL
;--------------------------------
;
WRTVRM: PUSH    AF              ;salva o byte
        LD      A,L             ;prepara o VDP para gravar
        OUT     (99H),A         ;um byte no endereco da VRAM
        LD      A,H             ;apontado por HL
        AND     3FH
        OR      40H
        OUT     (99H),A
        CALL    TEMPO           ;espera tempo para o VDP
        POP     AF              ;recupear o byte
        OUT     (98H),A         ;grava o byte
        RET                     ;retorna
;
;--------------------------------
; Tempo para o VDP
;--------------------------------
;
TEMPO:  EX      (SP),HL
        EX      (SP),HL
        RET
;
;--------------------------------
; Variaveis do programa
;--------------------------------
;
CONTAD: DEFB    0               ;contador de interrupcoes
VELOCI: DEFB    5               ;numero de interrupcoes para inverter o cursor
;
;--------------------------------
;
        END
```

NOVOS CARACTERES EM SCREEN 1

SILVIO CHAN

Tem havido um grande aumento no número de programas, principalmente jogos, desenvolvidos exclusivamente em SCREEN 1. Isso se deve `as facilidades oferecidas por essa tela, que antes era renegada a um plano inferior, como a possibilidade de se redefinir totalmente o conjunto de caracteres e de colorir cada um dos octetos (grupos de oito caracteres) com cores distintas. Além disso, a morosidade da tela SCREEN 2 em relação à produção de gráficos mais sofisticados através do BASIC MSX também contribui para a afirmação da SCREEN 1.

Este artigo é uma continuação do "PROGRAMAÇÃO GRÁFICA EM SCREEN 1", que foi publicado na revista CPU número 2.

O meu objetivo aqui é demonstrar aos leitores que com uma simples rotina em BASIC é possível alterar o desenho dos caracteres da SCREEN 1 para padrões mais compatíveis com os programas onde serão utilizados. Embora a alteração dos padrões dos caracteres possa ser feita através do programa redefinidor apresentado no artigo anterior, com a vantagem de podermos redefinir os caracteres para quaisquer novos padrões, o método de redefinição por rotina tem, por sua vez, a vantagem de ocupar apenas a memória necessária para armazenar a rotina, enquanto o redefinidor precisa de memória para guardar todo o novo conjunto de caracteres, isto é, dois Kbytes. Logo, a vantagem da rotina redefinidora é muito bem vista por quem faz programas extensos e precisa de alterar o padrão dos caracteres.

Para se criar uma rotina é preciso, além de conhecer a estrutura da VRAM, também ter conhecimentos sobre as tabelas verdade das operações lógicas do MSX (veja artigo publicado em CPU número 6), principalmente OR e AND e saber realizar rotações de bytes.

Partindo do pressuposto de que os leitores já possuem um certo conhecimento a respeito da estrutura da VRAM, vou apenas lembrar que a tabela de padrões dos caracteres em SCREEN 1 inicia e termina, originalmente, em 0 (zero) e 2047, respectivamente, tendo, portanto, a extensão de dois Kbytes.

As tabelas verdade das operações lógicas OR e AND são as seguintes:

TABELA VERDADE - OPERAÇÃO OR

A	OR	B	= C
0		0	0
1		0	1
0		1	1
1		1	1

TABELA VERDADE - OPERAÇÃO AND

A	AND	B	= C
0		0	0
1		0	0
0		1	0
1		1	1

A rotação de bytes é feita através da divisão ou da multiplicação do valor do byte por 2^n, onde n é o número de bits da rotação. Para fazer rotação de byte para a direita, deve-se dividir, e, para a esquerda, multiplica-se. Veja como:

ROTAÇÃO PARA A DIREITA
10000000 / 21 = 01000000
10000000 / 22 = 00100000
10000000 / 27 = 0000001

ROTAÇÃO PARA A ESQUERDA
00000001 * 21 = 00000010
00000001 * 22 = 0000100
00000001 * 27 = 10000000

A seguir, apresento oito exemplos de rotinas redefinidoras de padrões de caracteres e explico seus algoritmos.

CARACTERES BOLD

O conjunto de caracteres BOLD possui o traço mais largo e grosso do que o dos caracteres normais. Podemos obtê-lo através da rotação de um bit para a direita dos bytes que compõem os caracteres, que equivale a dividir o valor original por 2 e, em seguida, efetuando a operação OR entre o byte original e o alterado. Observe:

ROTAÇÃO OP.OR

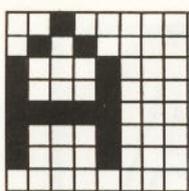
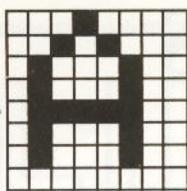
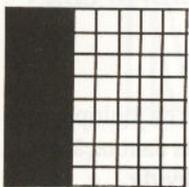
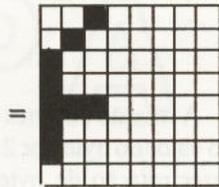
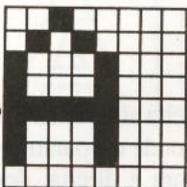
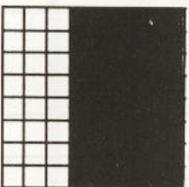
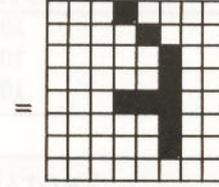
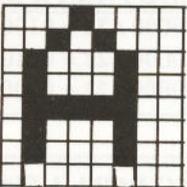
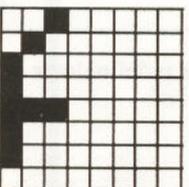
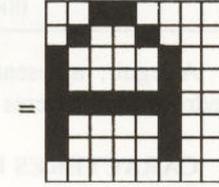
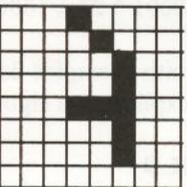
ORIGINAL ALTERADA BOLD

ROTINA PARA GERAR CARACTERES BOLD

```
10 SCREEN 1 : FOR I=1 TO 2047 : VPOKEI,VPEEK(I)
OR VPEEK(I)/2:NEXT
```

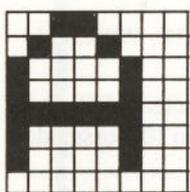
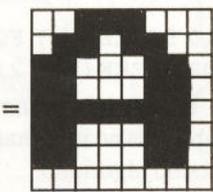
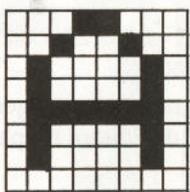
Observação: Uma outra rotina para gerar caracteres BOLD foi publicada no informativo da Editora Aleph número 10, possuindo o mesmo algoritmo.

CARACTERES LARGOS

Para se gerar caracteres LARGOS podemos usar o mesmo princípio gerador dos BOLD. No entanto, antes de realizar a operação OR entre a matriz alterada, será necessário colocar uma máscara através da operação AND sobre cada matriz. Na original serão preservados os três bits da esquerda e, na alterada, os cinco bits da direita. Só então será feita a operação OR entre as matrizes "mascaradas", obtendo, assim, os caracteres LARGOS.

ROTINA PARA GERAR CARACTERES LARGOS

```
10 SCREEN 1 : FOR I=0 TO 2047 : K=VPEEK(I) AND
&B11100000 : L=(VPEEK(I)/2) AND &B00011111 :
VPOKEI,K OR L : NEXT
```

CARACTERES LARGOS BOLD

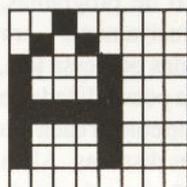
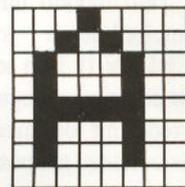
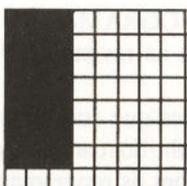
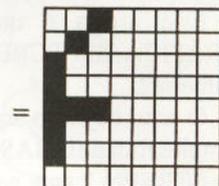
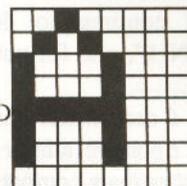
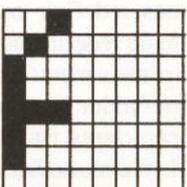
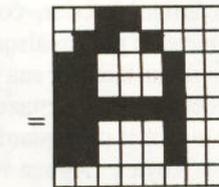
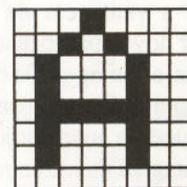
Este tipo de caracteres é o resultado da união das rotinas LARGO e BOLD. Para obtê-los, gera-se os caracteres LARGOS normalmente e, em seguida, realiza-se a operação OR entre a matriz larga original e uma matriz larga com um bit de rotação para a direita.

ROTINA PARA GERAR CARACTERES LARGOS BOLD

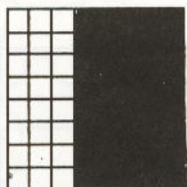
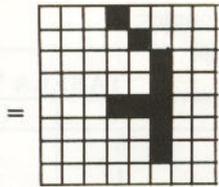
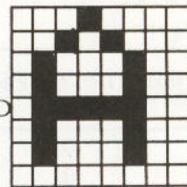
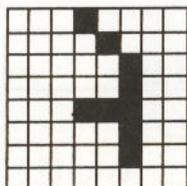
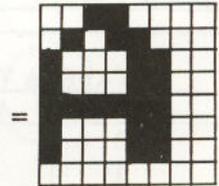
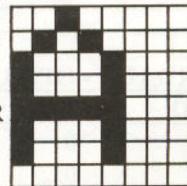
```
10 SCREEN 1 : FOR I=0 TO 2047 : K=VPEEK(I) AND
&B11100000 : L=VPEEK(I)/2) AND &B00011111 :
M=K OR L : VPOKE I,M OR M/2 : NEXT
```

CARACTERES SEMI-BOLD

Denominei estes caracteres de semi-bold porque, dos oito bits que formam um byte de padrão, quatro têm características BOLD e os outros quatro são comuns. Podem ser de dois tipos: ESQUERDA BOLD E DIREITA BOLD.

ESQUERDA BOLD

Para se obter este tipo, faz-se a rotação de um bit para a direita dos bytes da matriz original. Coloca-se uma máscara na original para que somente restem as três colunas da esquerda. Realiza-se a operação OR entre a matriz e a "mascarada", obtendo o novo caracter.

DIREITA BOLD

Seguindo o mesmo processo, consegue-se o tipo DIREITA BOLD. A única alteração será o uso da máscara na matriz alterada, fazendo com que restem somente as cinco colunas da direita, enquanto a matriz original não recebe nenhuma máscara, ficando inalterada.

ROTINAS PARA GERAR CARACTERES SEMI-BOLD

ESQUERDA BOLD

```
10 SCREEN 1 : FOR I=0 TO 2047 : K=VPEEK(I) AND
&B11100000 : L=VPEEK(I)/2 : VPOKEI,K OR L :
NEXT
```

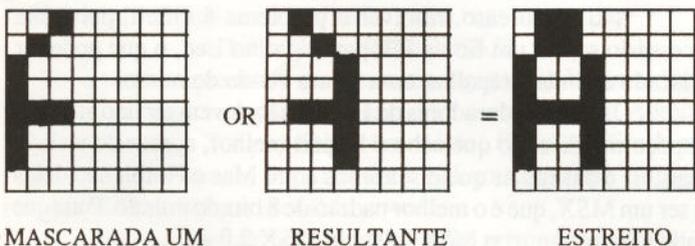
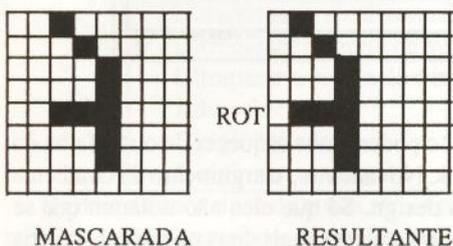
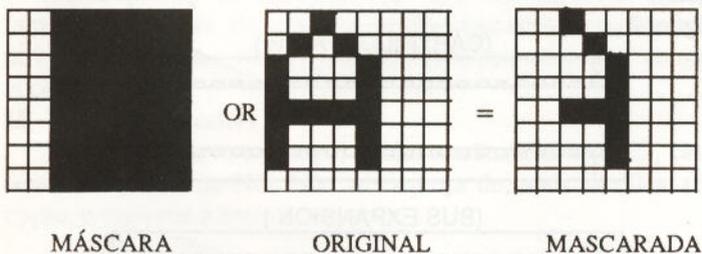
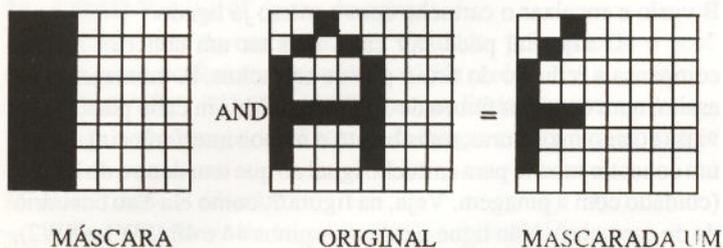
DIREITA BOLD

```
10 SCREEN 1 : FOR I=0 TO 2047 : K=(VPEEK(I)/2)
AND &B00011111 : L=VPEEK(I) : VPOKE I,KOR L :
NEXT
```

CARACTERES ESTREITOS

Para obter este conjunto, colocamos uma máscara sobre a matriz original de modo que sobrem apenas as três colunas da esquerda. O resultado será chamado de "mascarado um". Novamente, com a matriz original, coloca-se uma máscara com a operação AND, para que fiquem somente as seis colunas da direita. Multiplica-se o valor de cada byte por dois para conseguir uma rotação de um bit para a esquerda. Fazemos uma OR entre a resultante e a "mascarada um" e obtemos a matriz estreita.

ROTINA PARA GERAR CARACTERES ESTREITOS

```
10 SCREEN 1 : FOR I=0 TO2047 : K=VPEEK(I) AND
&B11100000 : L=(VPEEK(I) AND &B001111111)*2
:VPOKEI,K OR L: NEXT
```

CARACTERES INCLINADOS

Para se criar caracteres inclinados, basta fazer a rotação dos bytes da matriz dos caracteres de modo que os bytes de cima ou os de baixo recebam um maior deslocamento. Dessa maneira, os caracteres poderão aparecer inclinados para a direita ou para a esquerda.

INCLINADOS PARA A DIREITA (ITÁLICOS)

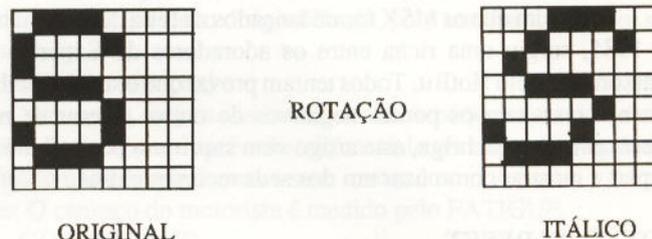
Nos caracteres inclinados para a direita ou itálicos, os bytes da matriz sofrem as seguintes rotações:

Primeiro e segundo bytes - dois bits para direita

Terceiro ao quinto - um bit para a direita

Sexto ao oitavo - inalterados.

ROTAÇÃO

ORIGINAL — ITÁLICO

INCLINADOS PARA A ESQUERDA

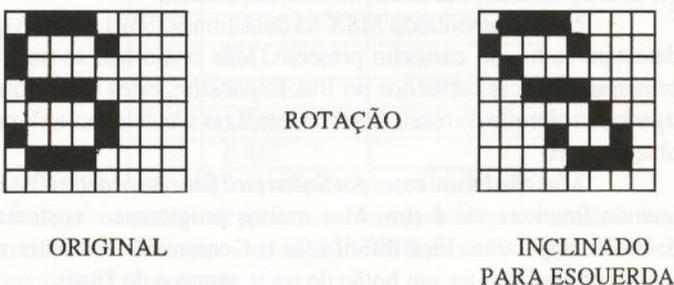
Nos caracteres inclinados para a esquerda, as rotações ocorrem de maneira inversa. Veja:

Primeiro e sgundo bytes - inalterados

Terceiro ao quinto - um bit para a direita

Sexto ao oitavo- dois bits para a direita

ROTAÇÃO

ORIGINAL — INCLINADO PARA ESQUERDA

ROTINAS PARA GERAR CARACTERES INCLINADOS

INCLINADOS PARA A DIREITA

```
10 SCREEN 1 : FOR I=0 TO 2047 STEP 8 : FOR N=1 TO
I+1 : VPOKENMV PEEK(N)/4 : NEXT : FOR N=I+2 TO I
+4 : VPOKEN, V PEEK(N)/2 : NEXT : NEXT
```

INCLINADOS PARA A ESQUERDA

```
10 SCREEN 1 : FOR I=0 TO 2047 STEP 8 : FOR N=I+2 T
O I+4 : VPOKE N, V PEEK(N)/2 : NEXT : FOR N=I+5 TOI
+7 : VPOKEN, V PEEK(N)/4 : NEXT : NEXT
```

BUS EXPANSION E RESET NO EXPERT

RUBENS HENRIQUE KÜHL JR.
PAULISOFT INFORMÁTICA

Assim que os MSX foram lançados na feira de informática de 1985, surgiu uma richa entre os adoradores do Expert e os apaixonados pelo HotBit. Todos tentam provar que o seu é o melhor micro e ressaltam os pontos negativos do outro. Apesar de não querer entrar nessa briga, este artigo vem suprir um ponto falho do Expert e mostrar como usar um dos seus recursos extras.

BOTÃO DE RESET

Sem sombra de dúvida, um grande erro da Gradiente foi não ter dotado o Expert de botão de reset. Ele existe em quase todos os outros MSX do mundo e o motivo por que ele foi excluído do Expert é estúpido: falta de lugar. A equipe de design não achou onde colocá-lo no gabinete, e a engenharia afirma que é impossível passar os fios pelo cabo do teclado.

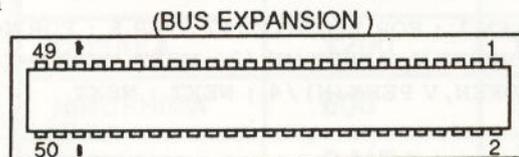
Mas o reset não foi esquecido. Há dentro do Expert um circuito completo de reset, incluindo supressor de oscilação da chave e temporizador para manter o sinal o tempo necessário. E onde ele está conectado? Ali atrás, no Bus Expansion.

No barramento do MSX há duas linhas (SW1 e SW2) que detectam se há um cartucho presente. Mas como não se pode (a princípio) colocar cartuchos no Bus Expansion, estes pinos foram ligados ao circuito de reset. Assim, basta ligar um deles ao +5V para obter o reset.

Mas não há um reset por Software? Ele não é melhor? Bem, quando funciona ele é sim. Mas muitos programas costumam deixá-lo inoperante. Fica difícil usar o Konami GameMaster nos jogos comuns sem ter um botão de reset, como o do HotBit.

Este botão de reset estraga o micro? Não, de forma alguma. Infelizmente, alguém divulgou um circuito de reset que usava o sinal RESET do barramento, esquecendo que esse sinal é exclusivamente de saída. O mesmo vale para alguns cartuchos que possuem este botão, já que os sinais SW1 e SW2 dos conectores A e B funcionam de modo normal.

A instalação é simples. Não é necessário abrir o micro, nem requer conhecimentos de eletrônica, apenas cuidado e atenção. Como se vê na figura 1, basta achar o Bus Expansion, contar a terceira fileira da esquerda para a direita e soldar (ou prender) dois fios ligados aos terminais de um interruptor (chave liga-desliga não serve, mas serve até botão de campainha!) e pronto.

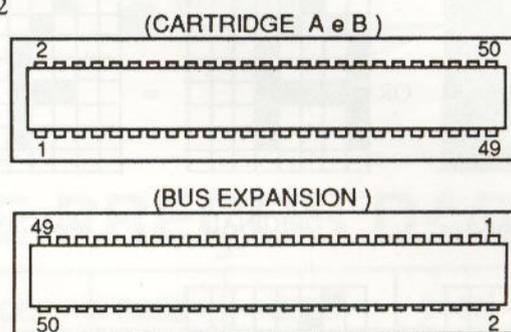
Figura 1

BUS EXPANSION

Até hoje, poucas pessoas vêem utilidade no Bus Expansion. A idéia da Gradiente era reduzir o problema do baixo número inicial de slots do MSX, fora o fato de que o Bus Expansion facilita a aplicação do Expert em controle industrial.

O Bus Expansion é uma extensão do conector frontal B, mas com uma limitação: se houver um cartucho no slot B, nenhum dispositivo com acesso por endereços (cartuchos com ROM ou com RAM entre outros) será selecionado.

Mas dispositivos controlados por portas, como as interfaces de comunicação e alguns programadores de EPROM, entre outros, podem ser ali ligados, deixando um slot livre. Outra aplicação é a cópia de programas em cartucho, bastando deixar o slot B vazio e encaixar o cartucho com o micro já ligado.

O material pode sair caro, mas ter um conector a mais compensa a redução do tira-e-põe de cartuchos. A montagem fica assim: num conector fêmea de 50 pinos, solde um cabo plano de 50 vias (quanto mais curto, mais barato e menos interferência) e neste um conector macho para cartucho igual ao que tem dentro do Expert (cuidado com a pinagem. Veja, na figura 2, como ela é ao contrário da do cartucho). Não ligue o cabo nos pinos 44 e 46 (SW1 e SW2), pois fazem parte do circuito de reset (a ligação do botão de reset fica igual).

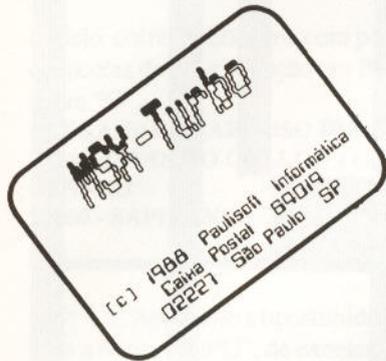
Figura 2

CAPS LOCK

Para finalizar, não poderíamos esquecer de outra falha do Expert: o led do Caps Lock. Novamente, o argumento da Gradiente é o cabo do teclado ou o design. Só que eles não notaram que se houvesse mais duas vias para o reset e mais duas para o led, haveria cabo e conector disponíveis até em loja de eletrônica.

No nosso caso, resolver o problema é difícil, pois seria necessário soldar um fio na PPI para ligar no Led, o que pode ser arriscado e ainda atrapalhar uma futura venda do micro.

Já que os adoradores do HotBit não devem ter lido até aqui, despeço-me dizendo que acho o Expert melhor, apesar do teclado (blargh!) e das falhas que discutimos aqui. Mas o Hotbit não deixa de ser um MSX, que é o melhor padrão de 8 bits do mundo. Pena que ainda não tenhamos o MSX 2.0 ou o MSX 2.0 +.

Rubens Henrique Kühl Jr. cursa engenharia elétrica na USP. Programa em MSX há três anos, trabalhando atualmente para a Paulisoft, onde desenvolve projetos em Basic, Pascal, C e Assembly, dentre eles o copiador de discos.

PAULISOFT Informática

DIGITE O PROGRAMA ABAIXO LEVA, APENAS 42 MINUTOS P/ RODAR

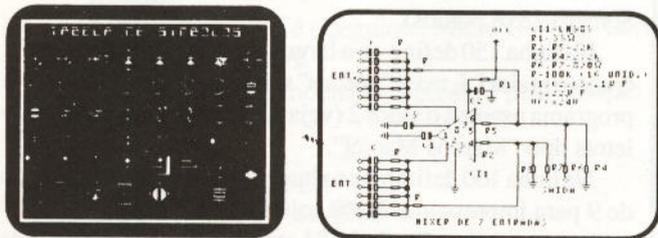
```
10 KEYOFF:COLOR15,4 ,4:SCREEN2
20 FORB=0TO127:X4=B*B:M=-128:A=SQR(16384
-X4):FORI=-ATOA:R=SQR(X4+I*I)/128:F=COS(
32*R)*(1-R)*4:Y=I/5+F*32:IFY<=MTHEN30ELS
EM=Y:Y=128+Y:X=128+B:PSET(X,280-Y):X=128
-B:PSET(X,280-Y)
30 IFINKEY$=""THENNEXTI:NEXTB
40 IFINKEY$=""THEN40
50 END
```

MAS COM O **MSXTURBO** INSTALADO, A EXECUÇÃO LEVA, APENAS 4 MINUTOS.

NÃO É MÁGICA É TECNOLOGIA

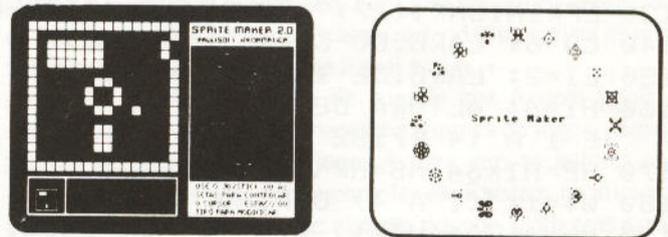
UM INCRÍVEL SOFTWARE QUE VAI DEIXAR SUAS ROTINAS DE CÁLCULO E PLOTAGEM DE GRÁFICOS DE 6 A 20 VEZES MAIS RÁPIDAS! MSX TURBO É UM COMPILADOR QUE OPERA NA MEMÓRIA, ACELERANDO INCRIVELMENTE AS OPERAÇÕES DE CÁLCULO.

FITA/DISCO - NCZ$ 25,80

Finalmente alguém pensou em você, técnico ou hobista de eletrônica e criou um auxiliar para seus projetos.
Tabela padrão de simbologia em Eletrônica;
Recursos p/edição, montagem e impressão de esquemas p/projetos eletrônicos.
Acompanha Arquivo Exemplo.

A partir de agora, a apresentação dos seus esquemas terá aquele toque profissional!

Autor: **PAULO MARQUES FIGUEIRA**

FITA/DISCO - NCZ$ 25,80

Super Editor de sprites 16x16 que inclui rotinas para reversão, espelho de 1/2 e 1/4.

O manual que acompanha o Software e genuinamente um curso sobre sprites. Inclui rotina para animar os sprites que você criar.

Ferramenta indispensável ao programador e a quem deseja explorar e conhecer os recursos do MSX.

Autor: **FÁBIO A. R. CORREA**

FITA/DISCO - NCZ$ 19,40

LANÇAMENTOS

GRAPHIC VIEW

Incrível programa para criar animações gráficas.

EROTIC SHOW - NCZ$ 6,00

MEGA MUSIC - NCZ$ 5,00

Software 100% nacional com manual, cópias com nº de série, garantia de "up to date" e assistência ao usuário.

Incríveis produtos que vão fazer o seu MSX virar uma máquina quente.

Envie seu pedido para a Cx. Postal 64.019 - CEP 02227 SP/SP, acompanhado de cheque nominal ou vale postal, ou visite-nos: Av. Prestes Maia 241 - Conj. 908 (a 100 metros da Estação São Bento do Metrô). Fone (011) 228-1313

Caso deseje solicite catálogo completo (grátis) c/ nossos Produtos.

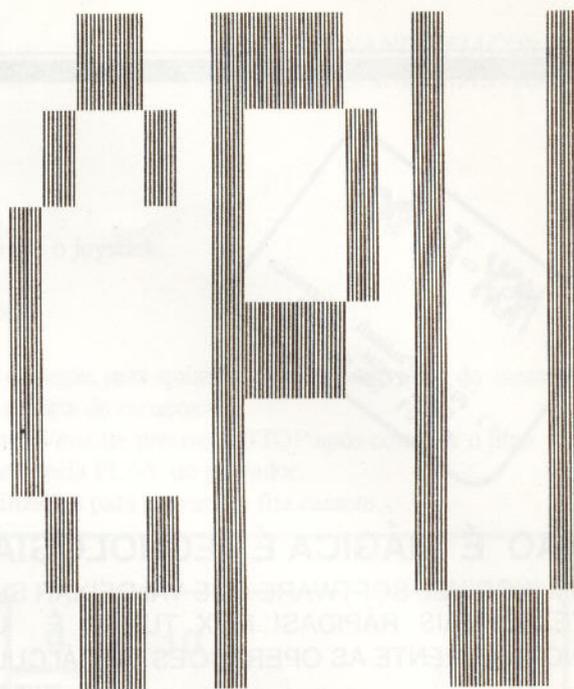
FAIXAS E CARACTERES NO MSX

PIERLUIGI PIAZZI
+50 DICAS PARA MSX
EDITORA ALEPH

Recentemente, meu filho de 13 anos me perguntou se o Apple da Editora estava livre para que ele pudesse usá-lo por algumas horas. Fiquei intrigado. Afinal de contas, minha editora é especializada em MSX e, entre Expert e Hotbit temos uma dezena dessas máquinas em uso. Porque ele haveria de querer logo um Apple?

Aí ele me explicou que queria fazer, na impressora, algumas faixas e cartazes para a feira de ciências do colégio, usando o programa "Printshop" do Apple.

Ofendidíssimo nos meus brios de "MSX-maníaco", sentei em frente a um Expert e, pouco tempo depois, gerei o programa listado nesta matéria.

Obviamente, como editor e escritor, resolvi aproveitá-lo, numa versão mais sofisticada, para o próximo lançamento da Aleph: "+50 Dicas para MSX". Mas, enquanto o livro não sai, o editor da CPU e eu resolvemos brindar os leitores desta revista com mais uma rotina útil para seu MSX.

```
100 'PROGRAMA DO LIVRO +50 DICAS P/MSX
110 'EDITORA ALEPH 1988
120 CLEAR1024:POKE&HF417,255:DEFINT A-Z
130 LPRINTCHR$(27)"A"CHR$(8);:SCREEN 1
140 LA=6: 'LARG.DO CARACT.(6/7/8)
150 LI=2: 'LARG.DE IMPR.(1 A 32)
160 HI=6: 'ALTURA DE IMPR.(1 A 9 P/80 COL
    E 1 A 14 P/132 COL)
170 HP=HI*64:MS=HP\256:LS=HP MOD 256
180 WT=1: '(1 A 4) DEFINE TEXTURA
190 WW=-170*(WT=1)-204*(WT=2)-240*(WT=3)
    -255*(WT=4)
200 INPUT"QUAL A MENSAGEM";A$:IF LEN(A$)
    >250 THEN CLS:PRINT"LONGA DEMAIS!":
    GOTO 200
210 GOSUB 350:FOR I=1 TO LEN(A$)
220 C$=MID$(A$,I,1):E=ASC(C$)*8
230 FORK=0TO7:X=E+K:Y$=BIN$(VPEEK(X))
240 B$(K)=RIGHT$("00000000"+Y$,8):NEXTK
250 FOR F=1TOLA:P$="":FOR J=7TO0STEP-1
260 P$=P$+MID$(B$(J),F,1):NEXT J
270 GOSUB 280:NEXT F,I:END
280 FOR M=1 TO LEN(P$):T$=MID$(P$,M,1):
    T(M)=VAL(T$)*WW:NEXT M
290 FOR N=1 TO LI
300 IF VAL("&B"+P$)=0 THEN 340
310 LPRINT CHR$(27)"K"CHR$(LS)CHR$(MS);
320 FOR M=1 TO LEN(P$):FOR G=1 TO 8*HI
330 LPRINT CHR$(T(M));:NEXT G,M
340 LPRINT CHR$(10);:NEXT N:RETURN
350 Q$="":FOR R=1 TO LEN(A$)
360 S$=MID$(A$,R,1):S=ASC(S$)
370 IF S>31 THEN Q$=Q$+S$
380 IF S=1 THEN Q$=Q$+CHR$
    (ASC(MID$(A$,R+1,1))-64):R=R+1
390 NEXT R:A$=Q$:RETURN
```

Após digitar (e gravar!) o programa listado, o usuário pode fazer algumas alterações para adptá-lo às suas necessidades.

Na linha 140 é definida a largura do caracter a ser usado (LA), que pode variar de 6 a 8. Usando o valor 6, teremos o espaçamento da SCREEN 0 (alguns caracteres gráficos sairão "cortados"). O valor 8 reproduz os caracteres como na Screen1 (8x8 pontos).

Na linha 150 define-se a largura de impressão (de 1 a 32). Quanto maior LI, mais "gordas" sairão as letras do cartaz. No programa usamos o valor 2 (veja a figura exemplo), gerando letras tipo "Marcos Maciel".

Na linha 160 definimos a altura das letras, até um máximo de 9 para impressoras de 80 colunas (cuidado: existem impressoras, como a Grafix MTA que aceitam um máximo de 7) ou 14 para 132 colunas.

Na linha 180, a variável WT define a textura da impressão, desde traços finos (WT=1) até traços cheios (WT=4)

Rode o programa algumas vezes com parâmetros diferentes e mensagens curtas, para se familiarizar com seu funcionamento.

Além da utilidade intrínseca da rotina, há uma lição que pode ser tirada deste programa: o MSX é um micro tão versátil e tão "USER FRIENDLY" que programadores medíocres (como é o meu caso), usando linguagens fáceis (como é o caso do Basic), podem gerar, rapidamente, rotinas que, em outra máquinas, exigiriam programadores de primeira linha, usando, no mínimo, linguagem assembly.

Esta talvez seja a maior virtude do MSX e a razão do seu sucesso.

Pierluigi Piazzi é autor de vários livros para MSX, sendo diretor- editor da Editora Aleph.

FORMATANDO ARQUIVOS DE TEXTO

GUSTAVO F. BAYER

Por causa do seu baixo custo e da facilidade de obtenção dos caracteres especiais da língua portuguesa no seu teclado, os micros MSX têm tudo para se tornarem o equipamento perfeito para a elaboração de textos. Por isso é quase inexplicável a inexistência de um programa mais completo de processamento de textos que, além dos recursos convencionais de definição de margens, salto automático de linhas com transferência de palavras excedentes e alinhamento à direita, manipulação de blocos etc., aproveite as facilidades do teclado MSX, seja no controle do cursor ou na geração dos caracteres acentuados e cedilhas do português.

Entre os processadores de texto disponíveis, o que mais se aproxima desse ideal é o SCED, normalmente usado na digitação de programas para posterior compilação.

Suas grandes vantagens são trabalhar em 80 ou 40 colunas de vídeo, com linhas de até 132 caracteres, aceitar todos os caracteres produzidos pelo teclado MSX e gravar os arquivos em formato ASCII puro (cada linha é gravada com os códigos ASCII de seus caracteres, seguidos por 0A e 0D, correspondentes ao "linefeed" e ao "return"). Isso permite a geração de arquivos facilmente tratados por rotinas de impressão já disponíveis, como a PIP.COM e até mesmo a incluída no MSXWRITE, ou por rotinas especificamente elaboradas para a impressora em uso, como as para a MTA e a LADY 80, já comentadas nessa revista. De resto, os demais comandos de deslocamento do cursor, "scroll", divisão e junção de linhas, busca e troca de palavras, gravação e leitura de arquivos e de pesquisa dos discos em uso, são no mínimo tão acessíveis e eficientes quanto nos outros processadores de texto.

Em alguns aspectos, entretanto, o SCED deixa a desejar.

Sua rotina de manipulação de blocos é trabalhosa: os blocos não são marcados, mas indicados por sua linha inicial e pelo número de linhas, e só podem ser copiados (sem opções de transferência ou eliminação). Mais grave, porém, é a impossibilidade de definição de margens e, por causa disso, a inexistência do salto automático de linha com transferência de palavra excedente e alinhamento à direita.

Para a ausência do salto automático de linha, a solução é proceder como na máquina de escrever: dando "return" ao fim de cada linha. Com isso, o alinhamento à direita tornar-se-ia extremamente trabalhoso, pois além do controle preciso do salto de linha, os espaços necessários teriam que ser inseridos manualmente entre as palavras.

O programa aqui apresentado foi desenvolvido exatamente para suprir a ausência de uma rotina de alinhamento automático no SCED, aproveitando a facilidade de manuseio dos arquivos por ele gravados. Em princípio, ele funciona de modo semelhante à função de acertar parágrafos dos editores de texto baseados no TASWORD: a partir da definição inicial do número desejado de colunas por linha, cada parágrafo do arquivo original é formatado com alinhamento automático à direita, e o texto formatado é gravado em um novo arquivo (cuja extensão é o número de colunas usado na formatação).

As situações interpretadas como início de novo parágrafo são as seguintes:

-linha em branco;
recuo de início de parágrafo (5 espaços vazios);
-linha iniciada com algarismo;
-linha iniciada com hífen ("sinal de menos").

Os trechos a serem excluídos da formatação, por exemplo tabelas, deverão ser precedidos e seguidos por linhas que contenham apenas dois caracteres ASCII 200 (GRAPH L no Hotbit ou LGraL no Expert).

O programa foi escrito em MBASIC, por esse ser o único BASIC acessível ao cartucho de 80 colunas da SHARP, e principalmente pela possibilidade de sua transformação em um programa com a extensão .COM, através do BASCOM, o que permite uma substancial aceleração do seu processamento. Para rodar em telas de 40 colunas, deverá ser acrescentada a seguinte linha:

```
75 WIDTH 40
```

A compilação através do **BASCOM** é bastante simples. Uma vez digitado e testado o programa, ele deve ser gravado no formato ASC II, através do comando:

<save "alinha.bas",a>.

Retornando ao DOS, comande:

<bascom alinha=alinha>.

O BASCOM analisará o programa, gravando um arquivo intermediário com o nome ALINHA.REL, encerrando com a mensagem de possíveis erros assinalados. Havendo erros, eles deverão ser corrigidos e o processamento pelo BASCOM deverá ser reiniciado. A seguir, comande:

<link alinha,alinha/n/e>,

e o BASCOM compilará e gravará o programa ALINHA.COM. (Para compilar outros programas, basta por o nome principal desse programa no lugar do "alinha" dos comandos acima citados.) Para facilitar a adaptação do programa em BASIC MSX por quem não disponha do BASCOM, ele foi escrito sem aproveitar recursos exclusivos do MBASIC, tais como os laços WHILE - WEND e a designação de variáveis por extenso. Por isso, basta digitar as seguintes linhas:

```
75 CLEAR 500:MAXFILES=2:KEYOFF
110 OPEN EN$ FORINPUTAS#1
190 OPEN SA$ FOROUTPUTAS#2
```

O programa em BASIC MSX é, entretanto, bastante mais lento que o programa compilado: apesar do tempo gasto com o acesso contínuo ao Drive ser o mesmo, o programa em BASIC MSX leva quase o dobro do tempo usado pelo programa COM no processamento de um arquivo de texto.

Como observação final, é importante registrar que esse programa não tem maiores pretensões que ser um "quebra-galho". Afinal, ele foi desenvolvido por um mero usuário levado ao autodidatismo na programação pela insuficiência de utilitários para o processamento de textos. Sugestões e programas mais eficientes são benvindos, principalmente se algum defensor do "software nacional" desenvolver, finalmente, um editor de textos à altura dos micros MSX aqui fabricados.

```
10 '*************************************
20 '* Formatador de arquivos de texto *
30 '*        Linguagem: MBASIC          *
40 '*************************************
50 '
60 '
70 ' *********** INICIALIZACAO *********
80 DIM PL$(40),TM(40)
90 PRINT CHR$(12):FOR F=1 TO 18:PRINT "
* ";:NEXT F:PRINT:PRINT"* FORMATADOR DE
 ARQUIVOS DE TEXTO *":FOR F=1 TO 18:PRI
NT"* ";:NEXT F:PRINT:PRINT:FILES:PRINT
CHR$(7)
100 PRINT CHR$(7):LINE INPUT "Arquivo a
 formatar ? ";EN$:IF EN$="" THEN GOTO 1
00
110 OPEN"I",#1,EN$
120 A=A+1:A$=MID$(EN$,A,1)
130 IF A$="" THEN SA$=EN$:GOTO 160
140 IF A$<>"." THEN GOTO 120
150 SA=A-1:SA$=LEFT$(EN$,SA)
160 PRINT CHR$(12):LINE INPUT"Alinhamen
to (direita) na coluna? ";CL$:IF CL$=""
 THEN GOTO 160
170 CL=VAL(CL$)
180 SA$=SA$+"."+CL$
190 OPEN"O",#2,SA$
200 PRINT CHR$(12)
210 ' **** VERIFICACAO DO ARQUIVO *****
220 IF EOF(1)<>-1 THEN LINE INPUT#1,LA$
 ELSE END
230 IF EOF(1)<>-1 THEN LINE INPUT#1,LC$
 ELSE LC$=""
240 FP=0:IF LC$="" OR LC$=CHR$(200)+CHR
$(200) OR LEFT$(LC$,5)="     " OR LEFT$
(LC$,1)=CHR$(45) OR (LEFT$(LC$,1)>CHR$(
47) AND LEFT$(LC$,1)<CHR$(58)) THEN FP=
1
250 TA=LEN(LA$)
260 IF LA$="" THEN GOTO 540 ELSE IF LA$
=CHR$(200)+CHR$(200) THEN GOTO 710
270 IF TA=CL THEN GOTO 540
280 IF TA<CL AND FP=1 THEN GOTO 540
290 IF TA<CL THEN GOTO 620
300 ' ****** PALAVRAS EXCEDENTES ******
310 CA=LEN(LA$):PL$=""
320 CA$=MID$(LA$,CA,1)
330 IF CA$>CHR$(32) THEN PL$=CA$+PL$:CA
=CA-1:GOTO 320
340 LB$=PL$+CHR$(32)+LB$
350 IF RIGHT$(LB$,1)=" " THEN LB$=LEFT$
(LB$,(LEN(LB$)-1)):GOTO 350
360 LA$=LEFT$(LA$,(CA-1)):TA=LEN(LA$)
370 IF LEN(LA$)>CL THEN GOTO 310
380 IF LEN(LA$)=CL THEN GOTO 540
390 ' **** DECOMPOSICAO DE LINHAS *****
400 FOR F=1 TO 40:PL$(F)="":TM(F)=0:NEX
T F
410 PL=0:CA=0
420 PL=PL+1:A$=PL$(PL-1):IF A$=" " OR A
$="  " OR A$="   " OR A$="    " OR A$="
     " THEN PL=PL-1
430 CA=CA+1:CA$=MID$(LA$,CA,1):PL$(PL)=
PL$(PL)+CA$
440 TM(PL)=TM(PL)+1
450 IF CA$>CHR$(32) THEN GOTO 430 ELSE
IF CA$=CHR$(32) THEN GOTO 420
460 ' *********** ALINHAMENTO **********
470 CP=CL-TA:IF CP=0 THEN GOTO 510
480 PA=PL
490 FOR F=1 TO CP:PL$(PA)=CHR$(32)+PL$(
PA):PA=PA-1:IF PA=1 THEN PA=PL
500 NEXT F
510 LA$=""
520 FOR F=1 TO PL:LA$=LA$+PL$(F):PL$(F)
="":NEXT F
530 ' ****** GRAVACAO E REINICIO ******
540 PRINT LA$:PRINT#2,LA$
550 IF LB$>"" AND FP=1 THEN LA$=LB$:LB$
="":GOTO 250
560 IF LEN(LB$)=>CL THEN LA$=LB$:LB$=""
:GOTO 250
570 IF LB$>"" THEN LA$=LB$+" "+LC$
580 IF LB$="" THEN LA$=LC$
590 LB$="":LC$="":IF EOF(1)<>-1 THEN GO
TO 230
600 IF LA$>"" THEN TA=LEN(LA$):GOTO 240
610 CLOSE:FOR F=1 TO 17:PRINT"* ";:NEXT
 F:PRINT:PRINT"* FORMATACAO DE TEXTO CO
NCLUIDA *":FOR F=1 TO 17:PRINT"* ";:NEX
T F:END
620 ' *** COMPLEMENTACAO DE LINHAS ****
630 IF LC$>"" THEN CA=0:PL$="":TC=0 ELS
E GOTO 540
640 CA=CA+1:CA$=MID$(LC$,CA,1):PL$=PL$+
CA$
650 IF CA$>CHR$(32) THEN TC=TC+1:GOTO 6
40
660 IF CA$=CHR$(32) THEN PL$=CHR$(32)+L
EFT$(PL$,(LEN(PL$)-1))
670 IF CA$="" THEN PL$=CHR$(32)+PL$
680 IF TA+1+TC>CL THEN GOTO 390
690 IF TA+1+TC<=CL THEN LA$=LA$+PL$:TA=
TA+1+TC:IF LEN(LC$)>TC+1 THEN LC$=RIGHT
$(LC$,(LEN(LC$)-(TC+1))) ELSE IF EOF(1)
<>-1 THEN LINE INPUT#1,LC$
700 IF TA=CL THEN GOTO 540 ELSE GOTO 63
0
710 ' ***** LINHAS SEM FORMATACAO *****
720 LA$=LC$:GOTO 750
730 IF EOF(1)<>-1 THEN LINE INPUT#1,LA$
 ELSE GOTO 610
740 IF LA$=CHR$(200)+CHR$(200) THEN GOT
O 220
750 PRINT LA$:PRINT#2,LA$:GOTO 730
```

SINTEVOZ

JÚLIO VELLOSO

Lendo artigos sobre a sintetização da voz, resolvi fazer um sintetizador para o MSX. Este programa é baseado na técnica de digitalização da voz, que converte um sinal analógico em digital, guardando-o na memória em formato binário para, em seguida, fazer o inverso, reproduzindo a voz.

O PROGRAMA

Utilizando o processador de som e o PPI, o programa funciona da seguinte forma:

ROTINA 1

Libera a RAM adicional (32k de memória) através da porta A8h da PPI. Liga o gravador, ressetando o bit 4 da porta AAh do mesmo.

Começa, então, a parte mais importante do programa. HL é iniciado com zero, 'DE' com a quantidade de bytes a serem gravados. Lê-se, no bit 7 da porta A2h a informação vinda do gravador, que pode ser 0 ou 1. Dependendo disso, o programa seta ou não o bit 7 do registrador C.

Depois, o programa faz uma rotação no mesmo, repetindo isto 8 vezes, até completar um byte, que é colocado na memória apontada por 'HL'. O apontador HL é incrementado e o DE decrementado, fazendo isto até que 'DE' fique igual a zero.

Por fim, a rotina desabilita a RAM adicional, desliga o gravador e volta ao BASIC.

ROTINA 2

O funcionamento desta outra rotina é o inverso da primeira, pegando um bit de um byte da memória apontada por 'HL' e joga-o para o processador de som, repetindo isto 8 vezes para cada byte e depois até que 32k bytes sejam enviados para o procesador de som.

Fazendo isto, o micro reproduz a voz anteriormente gravada.

OPERAÇÃO

Para fazer com que o programa funcione, digite a listagem do programa, tomando cuidado com os códigos em hexadecimal. Grave o programa e digite 'RUN'.
Ligue o gravador, pressionando:
'PLAY' + 'RECORD'.

Aperte espaço e fale ao microfone (o cabo do motor e o do load têm que estar conectados ao gravador).

Para que o micro reproduza a voz, digite '1'. Para outra gravação, digite '2' e para voltar ao BASIC '3'.

```
100 '              SINTEVOZ
110 '
120 '        Por: Julio Velloso
130 '
140 SCREEN0:COLOR15,1,1:KEYOFF:DEFUSR=&
HC000:DEFUSR1=&HC038
150 RESTORE280:FORA=0TO135:READA$:POKE&
HC000+A,VAL("&h"+A$):NEXT:A=INP(&HA8):I
FA=&HA0THENPOKE&HC008,&HAA:POKE&HC02D,&
HA0:POKE&HC03A,&HAA:POKE&HC07A,&HA0
160 PRINTSPC(15);"SINTEVOZ":PRINT:PRINT
SPC(10);"Por: Julio Velloso":PRINT:PRIN
T:PRINT
170 PRINT"- Coloque o plug do load e ti
re o do     save.":PRINT:PRINT"- Aperte
 'play' + 'record'."
180 PRINT:PRINT:PRINT:PRINT:PRINTSPC(12
);"APERTE ESPACO."
190 A=STRIG(0):IFA=0THEN190ELSEA=USR(0)
200 '
210 '            S P E A K
220 '
230 PRINT:PRINT:PRINTSPC(9);"1 - Reprod
uz a voz.":PRINT:PRINTSPC(9);"2 - Outra
 Gravacao.":PRINT:PRINTSPC(9);"3 - Volt
a ao BASIC."
240 A$=INKEY$:IFA$=""THEN240ELSEA=VAL(A
$):ON A GOSUB 250,260,270:GOTO 240
250 A=USR1(0):RETURN
260 SCREEN0:RETURN160
270 SCREEN0:KEYON:END
280 DATA F3,DB,AA,CB,A7,D3,AA,3E
290 DATA FF,D3,A8,21,00,00,11,FF
300 DATA 7F,0E,00,06,08,DB,A2,CB
310 DATA 7F,20,04,CB,F9,18,02,CB
320 DATA 89,CB,29,10,F2,71,1B,23
330 DATA 7A,B3,20,E7,3E,F0,D3,A8
340 DATA DB,AA,CB,E7,D3,AA,FB,C9
350 DATA F3,3E,FF,D3,A8,3E,07,D3
360 DATA A0,3E,08,D3,A1,3E,00,D3
370 DATA A0,3E,00,D3,A1,3E,01,D3
380 DATA A0,3E,00,D3,A0,3E,08,D3
390 DATA A0,21,00,00,11,FF,7F,4E
400 DATA 06,08,F3,CB,41,28,04,3E
410 DATA 0F,18,02,3E,00,D3,A1,CB
420 DATA 29,10,EF,23,1B,7A,B3,20
430 DATA E6,3E,F0,D3,A8,3E,08,D3
440 DATA A0,3E,00,D3,A1,FB,C9,00
```


Júlio Velloso é estudante da UCP, na qual cursa o 4º período do Curso de Ciência da Computação. Programa em Basic, Assembler Z80, Pascal e dBase III, há quatro anos, nos equipamentos IBM-PC e, principalmente, MSX. É autor do programa utilitário de cópia BKP, comercializado inicialmente pela Ciberne e do Sistema Operacional BKP Disco, comercializado pela Paulisoft Informática.

PROJETO MSXDEBUG PARTE I

SÉRGIO DURIC CALHEIROS

Frequentemente, ao abrir uma boa revista especializada em software, nos deparamos com bons programas, mas que, invariavelmente, estão contidos em listagens quilométricas em Basic, contendo código em dezenas de linhas DATA. Imagine o trabalho e o desconforto de digitá-las utilizando apenas o editor Basic durante horas seguidas. Isso sem falar do código inútil como a parte montadora e a própria sintaxe da linguagem. Mesmo assim, dizem, um bom programa vale o sacrifício! Será?

Que tal se todo esse trabalho fosse amenizado e esta fosse a última vez em que você tivesse que digitar partes dispensáveis? Que tal dispor de um editor rápido e versátil e que permita acessar qualquer parte da memória que o Basic não alcança? Que tal ter em mãos não só um editor, mas um sistema que possa ser expandido à medida que surgem as necessidades?

Amigo leitor, é exatamente isso que propomos neste artigo. O MSXDEBUG é um programa que foi idealizado vizando ser complementado aos poucos.

O programa consiste, basicamente, de meia dúzia de comandos que formam a base de um sistema integrado. Este sistema contém todas as rotinas necessárias à manipulação de arquivos em disco, rotinas de reconhecimento de comandos e várias outras sub-rotinas que serão úteis no futuro.

Inicialmente, estão prontas as rotinas que formam a base do MSXDEBUG. São rotinas de manipulação de memória como DISPlay, EXECutar, MOVer e FILL (preencher com dado valor). Além dessas quatro, também temos o DiskSAVE e o DiskLOAD (DSAVE e DLOAD, respectivamente) que permitem salvar um bloco da memória no disco e carregar QUALQUER tipo de arquivo que esteja no disco, seja ele executável, seja ele binário ou mesmo o próprio sistema operacional. O último comando implementado é o comando DIR que mostra o diretório do disco sem maiores detalhes.

A utilização do MSXDEBUG é bem fácil e acessível, principalmente àquele que está iniciando no assembler.

Após a digitação da listagem, salve o programa e rode-o.

Automaticamente, o seu MSXDEBUG será montado e, caso haja algum erro de digitação, será mostrada a linha em que ocorre o erro. Não esqueça de iniciar o programa como mostra a listagem, pois, se ocorrer erro, será indicada outra linha que não é a correta. Ao final do processamento, volte ao DOS e digite MSXDEBUG. A tela deverá conter a mensagem inicial e o prompt esperando que seja digitado alguma coisa.

A seguir está uma descrição mais detalhada do que faz cada comando.

1) **DISP** endereço
Mostra o conteúdo da memória dado por endereço. Para avançar ou retroceder, utilize as teclas dos cursores. Estando na posição desejada, pressione <CR>. Neste momento, você poderá alterar o conteúdo desta memória ou se deslocar ao longo dos códigos livremente. Para alterar utilizando diretamente o teclado, pressione <CR> novamente. Repita sempre que precisar mudar o modo de edição. Para sair, basta pressionar <ESC>.

2) **EXEC** endereço
Executa este endereço sem break-points.

3) **MOVE** início, fim, destino
Move um bloco na memória. Início deve ser menor que fim.

4) **FILL** início, fim, dado
Preenche uma área de memória especificado por início e fim, com o dado fornecido.

5) **DIR**
Mostra o diretório do disco de forma simples.

6) **DSAVE** nome. ext, início, fim
Salva o bloco limitado por início e fim no disco com o nome nome.ext. Salva qualquer coisa em qualquer lugar.

7) **DLOAD** nome.ext endereço
Carrega qualquer arquivo, desde que esteja no disco e o coloca no endereço dado por endereço. Nestes três comandos não é possível especificar o drive. Será utilizado aquele que estava habilitado antes de executar o MSXDEBUG.

8) **DOS**
Encerra as atividades no MSXDEBUG e retorna ao DOS.

As mensagens de erro do sistema são mostradas na hora devida. Para vê-las, experimente passear pela RAM do micro a partir do endereço100H, que é onde os programas com extensão COM são carregados. Examine também a área reservada para as variáveis do BASIC. Para examinar um programa qualquer, preencha uma área de memória com um dado valor (usualmente 0) e use o comando DLOAD. Você tem cerca de 50K livres ou mais, dependendo do número de drives utilizados. Aprenderemos como utilizá-los integralmente e também criar programas que os utilizem.

Infelizmente, esta versão só serve para micro computadores equipados com disco. Está sendo preparada uma versão que utiliza o cassete e que, breve, será publicada. Nos próximos números, começaremos a analisar as rotinas do MSXDEBUG e aprenderemos como implementar rotinas e a utilizar outras. Aguardem as novidades.

Sérgio Duric Calheiros é estudante da PUC-RJ, onde cursa Engenharia de Computação, desde 1985. Auto-didata em Basic e Assembler e algumas outras linguagens, programando em micros da linha MSX e PC.

```
100 CLS:KEY OFF
110 PRINT "MSXDEBUG VERSAO 1.0"
120 GERAL=0
130 PRINT"PROGRAMA SENDO CONVERTIDO!"
140 OPEN "MSXDEBUG.COM" AS #1 LEN=1
150 FIELD #1,1 AS A$
160 FOR I%=1 TO 384
170 LOCATE 0,4:PRINT "CONTAGEM:";I%
180 SOMA=0
190 FOR J%=1 TO 8
200 READ K$:K=VAL("&H"+K$):SOMA=SOMA+K
210 LSET A$=CHR$(K):PUT #1
220 NEXT J%
230 READ K
240 GERAL=GERAL+SOMA
250 IF K<>SOMA THEN PRINT "ERRO NA LINH
A";I%*10+290:END
260 NEXT I%
270 READ K
280 IF K<>GERAL THEN PRINT "ERRO NO PRO
GRAMA":END
290 CLOSE:END
300 DATA C3,89,07,CD,C9,08,CD,9A,1112
310 DATA 08,CD,FA,08,7D,2A,8B,0D,790
320 DATA ED,5B,89,0D,12,A7,ED,52,982
330 DATA 44,4D,0B,EB,54,5D,13,ED,824
340 DATA B0,C9,CD,C9,08,CD,9A,08,1158
350 DATA CD,FA,08,22,7D,0D,ED,5B,963
360 DATA 89,0D,CD,33,0B,38,1B,2A,542
370 DATA 8B,0D,ED,5B,89,0D,A7,ED,1034
380 DATA 52,44,4D,2A,89,0D,09,E5,657
390 DATA 2A,7D,0D,09,D1,EB,03,ED,873
400 DATA B8,C9,2A,8B,0D,ED,5B,89,1044
410 DATA 0D,A7,ED,52,44,4D,2A,7D,811
420 DATA 0D,03,EB,ED,B0,C9,CD,9A,1224
430 DATA 08,CD,FA,08,CD,72,01,C3,986
440 DATA C9,07,E9,CD,FA,0A,21,91,1084
450 DATA 0D,11,92,0D,01,24,00,36,280
460 DATA 20,ED,B0,CD,9A,08,2A,89,991
470 DATA 0D,CD,E9,08,22,83,0D,FD,890
480 DATA 21,15,0B,CD,74,F9,CD,0F,855
490 DATA 0B,CD,F2,01,CD,E8,0A,11,923
500 DATA 87,05,21,70,05,CD,61,02,594
510 DATA 30,F2,CD,AF,01,18,ED,E9,1165
520 DATA CD,18,0B,C3,0B,02,11,C8,665
530 DATA 00,CD,7A,02,11,65,05,CD,657
540 DATA 24,0B,CD,0B,02,CD,01,02,473
550 DATA 11,B8,00,CD,84,02,C9,11,758
560 DATA 80,01,CD,7A,02,3E,0B,CD,736
570 DATA 11,0B,C3,F2,01,3E,0B,CD,744
580 DATA 11,0B,C3,F2,01,CD,01,02,674
590 DATA FD,21,2B,0B,CD,74,F9,C3,1105
600 DATA C9,07,06,17,C5,CD,0B,02,652
610 DATA CD,18,0B,C1,10,F6,C3,0B,901
620 DATA 02,06,18,3E,1F,CD,11,0B,358
630 DATA 10,F9,C9,11,91,0D,2A,83,814
640 DATA 0D,22,85,0D,CD,09,08,13,434
650 DATA 06,08,DD,21,17,0E,C5,2A,544
660 DATA 83,0D,7E,32,85,0D,CD,50,751
670 DATA 02,DD,23,21,85,0D,CD,FB,893
680 DATA 07,CD,FB,07,13,2A,83,0D,675
690 DATA 23,22,83,0D,C1,10,DF,AF,820
700 DATA 12,DD,77,00,11,91,0D,CD,738
710 DATA 24,0B,11,17,0E,C3,24,0B,343
720 DATA DD,36,00,2E,FE,20,D8,FE,1077
730 DATA FF,C8,FE,7F,C8,DD,77,00,1376
740 DATA C9,06,00,4F,7E,A7,C8,04,783
750 DATA B9,23,20,F8,EB,78,2B,2B,941
760 DATA 23,23,10,FC,7E,23,66,6F,712
770 DATA 37,C9,2A,83,0D,A7,ED,52,928
780 DATA 22,83,0D,C9,2A,83,0D,19,590
790 DATA 22,83,0D,C9,2A,83,0D,22,599
800 DATA 7F,0D,CD,E9,02,CD,39,0B,853
810 DATA CD,E8,0A,CD,EC,04,11,93,1056
820 DATA 05,21,77,05,CD,61,02,DC,686
830 DATA AF,01,18,EC,2A,81,0D,11,637
840 DATA 08,00,A7,ED,52,22,81,0D,670
850 DATA C9,2A,81,0D,11,08,00,19,435
860 DATA 22,81,0D,C9,D1,2A,7F,0D,768
870 DATA 22,83,0D,CD,40,0B,C3,01,654
880 DATA 02,2A,81,0D,11,07,00,19,235
890 DATA ED,5B,83,0D,C3,33,0B,2A,771
900 DATA 81,0D,ED,5B,83,0D,C3,33,860
910 DATA 0B,2A,7F,0D,22,83,0D,11,388
920 DATA C0,00,CD,7A,02,22,81,0D,697
930 DATA 3E,0B,CD,11,0B,11,69,05,433
940 DATA C3,24,0B,CD,40,0B,11,69,644
950 DATA 05,CD,24,0B,06,08,11,7F,415
960 DATA 05,CD,24,0B,10,F8,CD,52,808
970 DATA 05,CD,39,0B,CD,E8,0A,CD,930
980 DATA 95,04,11,A1,05,21,77,05,493
990 DATA CD,61,02,DC,AF,01,18,EC,960
1000 DATA D1,D1,2A,7F,0D,22,83,0D,778
1010 DATA C3,40,0B,D1,CD,40,0B,C3,954
1020 DATA 32,05,2A,7F,0D,11,08,00,262
1030 DATA A7,ED,52,ED,5B,83,0D,EB,1193
1040 DATA C3,33,0B,2A,7F,0D,11,B9,641
1050 DATA 00,A7,ED,52,ED,5B,83,0D,958
1060 DATA C3,33,0B,CD,D1,02,28,11,730
1070 DATA CD,40,0B,11,01,00,CD,84,635
1080 DATA 02,3E,1C,CD,11,0B,C3,39,577
1090 DATA 0B,CD,40,0B,06,07,3E,1D,395
1100 DATA CD,11,0B,10,F9,11,07,00,522
1110 DATA CD,7A,02,CD,39,0B,CD,42,873
1120 DATA 03,DA,62,04,CD,76,04,11,667
1130 DATA 69,05,CD,24,0B,06,08,11,393
1140 DATA 7F,05,CD,24,0B,10,F8,C3,843
1150 DATA 39,0B,CD,DF,02,28,11,CD,760
1160 DATA 40,0B,11,01,00,CD,7A,02,422
1170 DATA 3E,1D,CD,11,0B,C3,39,0B,587
1180 DATA CD,40,0B,06,07,3E,1C,CD,588
1190 DATA 11,0B,10,F9,11,07,00,CD,522
1200 DATA 84,02,CD,39,0B,C3,EA,03,839
1210 DATA CD,42,03,DA,62,04,CD,76,917
1220 DATA 04,11,69,05,CD,24,0B,C3,578
1230 DATA 39,0B,CD,53,03,D0,CD,40,836
1240 DATA 0B,11,08,00,CD,7A,02,CD,570
1250 DATA AC,02,3E,1E,CD,11,0B,C3,694
1260 DATA 39,0B,CD,D1,02,28,12,CD,747
1270 DATA 40,0B,11,01,00,CD,84,02,432
1280 DATA 11,7F,05,CD,24,0B,C3,39,653
1290 DATA 0B,CD,40,0B,06,07,11,83,452
1300 DATA 05,CD,24,0B,10,F8,11,07,545
1310 DATA 00,CD,7A,02,CD,39,0B,C3,797
1320 DATA D8,03,CD,DF,02,28,12,CD,912
1330 DATA 40,0B,11,01,00,CD,7A,02,422
1340 DATA 11,83,05,CD,24,0B,C3,39,657
1350 DATA 0B,CD,40,0B,06,07,11,7F,448
1360 DATA 05,CD,24,0B,10,F8,11,07,545
1370 DATA 00,CD,84,02.CD.39.0B.C3.807
1380 DATA EA,03,CD,40,0B,11,08,00,542
1390 DATA CD,84,02,CD,B9,02,3E,1F,824
1400 DATA CD,11,0B,C3,39,0B,CD,40,765
1410 DATA 0B,CD,18,0B,2A,7F,0D,22,467
1420 DATA 83,0D,11,08,00,19,22,7F,355
1430 DATA 0D,CD,0B,02,11,08,00,CD,461
1440 DATA 7A,02,C3,B9,02,FE,20,D8,1008
1450 DATA F5,CD,40,0B,F1,2A,83,0D,952
1460 DATA 77,E5,2A,81,0D,22,83,0D,710
1470 DATA 3E,0D,CD,11,0B,CD,0B,02,526
1480 DATA E1,22,83,0D,3A,81,0D,6F,714
1490 DATA 3A,83,0D,95,6F,3E,08,95,681
1500 DATA 47,3E,1D,CD,11,0B,10,F9,660
1510 DATA CD,39,0B,3E,1C,C9,FE,30,866
1520 DATA D8,4F,3E,66,B9,79,D8,FE,1235
1530 DATA 61,38,02,D6,20,4F,3E,46,612
1540 DATA B9,79,D8,FE,61,38,03,FE,1186
1550 DATA 41,D8,A7,C9,CD,CE,04,D8,1280
1560 DATA 21,25,0E,77,F5,CD,40,0B,728
1570 DATA F1,CD,11,0B,CD,39,0B,CD,952
1580 DATA E8,0A,CD,CE,04,38,F8,21,994
1590 DATA 25,0E,23,77,23,36,00,F5,539
1600 DATA CD,40,0B,F1,CD,11,0B,CD,959
1610 DATA FA,08,7D,2A,83,0D,77,E5,917
1620 DATA 2A,81,0D,22,83,0D,3E,0D,437
1630 DATA CD,11,0B,CD,0B,02,E1,22,710
```

```
1640 DATA 83,0D,11,69,05,CD,24,0B,523
1650 DATA 3A,81,0D,6F,3A,83,0D,95,662
1660 DATA A7,28,09,47,11,7F,05,CD,641
1670 DATA 24,0B,10,F8,CD,39,0B,3E,646
1680 DATA 1C,C9,3A,81,0D,6F,3A,83,729
1690 DATA 0D,95,A7,C8,47,3E,1C,CD,895
1700 DATA 11,0B,10,F9,C9,0B,1B,4C,608
1710 DATA 00,0D,1C,1C,1C,1C,1C,00,153
1720 DATA 1B,0D,1F,1E,1D,1C,00,1B,185
1730 DATA 0D,1F,1E,1D,1C,08,00,1C,167
1740 DATA 1C,1C,00,1D,1D,1D,00,E5,372
1750 DATA 01,8C,02,B0,01,B6,01,CF,710
1760 DATA 01,DD,01,C4,02,03,03,D8,643
1770 DATA 03,EA,03,32,04,02,04,32,350
1780 DATA 04,30,03,3B,03,8E,03,EA,496
1790 DATA 03,AA,03,63,03,AA,03,21,484
1800 DATA 17,0E,06,0B,36,3F,23,10,222
1810 DATA FB,CD,66,07,CD,4A,07,11,868
1820 DATA BB,0D,0E,19,0C,CD,05,00,461
1830 DATA 11,5C,00,0E,11,CD,05,00,350
1840 DATA B7,11,B6,0B,C2,06,0B,11,621
1850 DATA 5C,00,CD,18,0B,06,03,AF,516
1860 DATA 32,C7,0D,11,BC,0D,CD,24,721
1870 DATA 0B,11,59,0C,CD,24,0B,0E,395
1880 DATA 12,C5,CD,05,00,C1,B7,C0,993
1890 DATA 10,E5,CD,18,0B,06,03,18,518
1900 DATA DE,CD,9A,08,CD,86,09,CD,1142
1910 DATA 66,07,CD,9A,08,CD,FA,08,939
1920 DATA 22,89,0D,11,5C,00,CD,4A,572
1930 DATA 07,CD,3F,06,11,45,0C,C2,573
1940 DATA 06,0B,CD,72,07,ED,5B,89,808
1950 DATA 0D,D5,0E,19,0C,CD,05,00,487
1960 DATA 11,5C,00,CD,48,06,D1,C0,793
1970 DATA 21,80,00,19,EB,18,EA,D5,892
1980 DATA 0E,0F,CD,05,00,B7,D1,C9,832
1990 DATA E5,D5,C5,0E,14,CD,05,00,883
2000 DATA FE,02,11,10,0C,CA,06,0B,520
2010 DATA C1,D1,E1,B7,C9,CD,B6,06,1404
2020 DATA CD,66,07,CD,4A,07,CD,3F,868
2030 DATA 06,20,03,CD,3E,07,11,5C,424
2040 DATA 00,CD,05,07,11,80,00,0E,376
2050 DATA 19,0C,C3,05,00,CD,A5,08,615
2060 DATA CD,66,07,CD,72,07,CD,5D,938
2070 DATA 06,2A,89,0D,ED,5B,8B,0D,678
2080 DATA CD,33,0B,30,1B,00,11,80,487
2090 DATA 00,01,80,00,ED,B0,11,5C,651
2100 DATA 00,CD,2B,07,2A,89,0D,11,464
2110 DATA 80,00,19,22,89,0D,18,D9,578
2120 DATA 11,5C,00,C3,18,07,2A,8B,516
2130 DATA 0D,ED,5B,89,0D,A7,ED,52,977
2140 DATA E5,CD,DB,06,CD,F6,06,D1,1325
2150 DATA 06,0A,CB,3A,CB,1B,10,FA,773
2160 DATA 13,CD,33,0B,D0,11,D3,0B,733
2170 DATA C3,06,0B,21,E1,0D,06,0B,500
2180 DATA 36,3F,23,10,FB,11,BB,0D,636
2190 DATA 0E,19,0C,CD,05,00,11,5C,370
2200 DATA 00,0E,11,C3,05,00,1E,00,261
2210 DATA 0E,1B,CD,05,00,FE,02,C8,707
2220 DATA CB,3C,CB,1D,C9,E5,D5,C5,1335
2230 DATA 0E,16,CD,05,00,3C,11,C3,518
2240 DATA 0B,CA,06,0B,C1,D1,E1,C9,1058
2250 DATA E5,D5,C5,0E,10,CD,05,00,879
2260 DATA B7,11,F3,0B,C2,06,0B,C1,858
2270 DATA D1,E1,C9,E5,D5,C5,0E,15,1309
2280 DATA CD,05,00,B7,11,20,0C,C2,648
2290 DATA 06,0B,C1,D1,E1,C9,E5,D5,1287
2300 DATA C5,0E,13,CD,05,00,C1,D1,842
2310 DATA E1,C9,21,EC,0D,E5,D1,13,1165
2320 DATA 36,00,01,18,00,ED,B0,21,525
2330 DATA E0,0D,11,5C,00,01,25,00,384
2340 DATA ED,B0,11,5C,00,C9,21,17,779
2350 DATA 0E,11,E1,0D,01,0B,00,ED,518
2360 DATA B0,C9,21,17,0E,11,05,0E,483
2370 DATA 01,08,00,ED,B0,3E,2E,12,548
2380 DATA 13,01,03,00,ED,B0,AF,12,629
2390 DATA C9,ED,7B,06,00,3E,28,32,719
2400 DATA B0,F3,21,91,0D,11,92,0D,786
2410 DATA 01,FF,00,36,00,ED,B0,21,756
2420 DATA 17,0E,54,5D,13,36,20,01,320
2430 DATA 0B,00,ED,B0,CD,72,07,AF,925
2440 DATA 32,A9,FC,ED,7B,06,00,CD,1042
2450 DATA FA,0A,CD,0F,0B,11,60,0C,616
2460 DATA CD,24,0B,CD,18,0B,C3,20,719
2470 DATA 08,ED,7B,06,00,CD,18,0B,614
2480 DATA C3,20,08,CD,FA,0A,FD,21,986
2490 DATA A0,F9,DD,21,3E,00,CD,74,1046
2500 DATA F9,CD,18,0B,C3,00,00,11,701
2510 DATA B6,0D,CD,09,08,AF,12,11,627
2520 DATA B6,0D,CD,24,0B,11,1B,08,499
2530 DATA C3,24,0B,AF,ED,6F,FE,0A,1029
2540 DATA 38,02,C6,07,C6,30,12,13,546
2550 DATA C9,21,85,0D,23,CD,FB,07,878
2560 DATA CD,FB,07,2B,CD,FB,07,C3,1164
2570 DATA FB,07,00,1D,1D,1D,1D,00,374
2580 DATA CD,18,0B,11,5C,0C,CD,24,602
2590 DATA 0B,CD,62,0A,CD,27,09,FE,831
2600 DATA 20,28,ED,A7,28,EA,CD,4E,1033
2610 DATA 08,78,21,82,0C,23,23,10,389
2620 DATA FC,2B,2B,7E,23,66,6F,CD,917
2630 DATA 4D,08,C3,C9,07,E9,22,87,890
2640 DATA 0D,ED,5B,87,0D,21,D2,0C,744
2650 DATA 06,01,EB,7E,EB,A7,28,24,846
2660 DATA FE,20,28,20,FE,61,38,02,767
2670 DATA D6,20,BE,23,13,28,EB,7E,891
2680 DATA A7,23,20,FB,04,7E,FE,FF,1124
2690 DATA ED,5B,87,0D,20,DC,11,31,794
2700 DATA 0C,C3,06,0B,7E,21,31,0C,444
2710 DATA EB,A7,C2,06,0B,EB,C5,21,1078
2720 DATA 25,0E,01,20,00,EB,ED,B0,732
2730 DATA C1,C9,CD,27,09,A7,C0,11,1023
2740 DATA 8A,0B,C3,06,0B,CD,9A,08,728
2750 DATA CD,86,09,CD,9A,08,CD,FA,1170
2760 DATA 08,22,89,0D,E5,CD,9A,08,788
2770 DATA CD,FA,08,22,8B,0D,D1,CD,1063
2780 DATA 33,0B,11,9E,0B,DA,06,0B,483
2790 DATA C9,CD,9A,08,CD,FA,08,22,1065
2800 DATA 89,0D,CD,9A,08,CD,FA,08,980
2810 DATA 22,8B,0D,ED,5B,89,0D,CD,869
2820 DATA 33,0B,D0,11,9E,0B,C3,06,657
2830 DATA 0B,C2,FA,08,3E,1D,CD,11,776
2840 DATA 0B,E5,22,85,0D,CD,E7,07,863
2850 DATA E1,C9,CD,27,09,11,8A,0B,845
2860 DATA CA,06,0B,CD,3B,09,21,00,525
2870 DATA 00,22,8D,0D,21,12,0E,7E,379
2880 DATA 7E,A7,28,23,FE,20,28,1F,725
2890 DATA CD,65,09,11,9E,0B,DA,06,725
2900 DATA 0B,CD,7B,09,23,18,E8,21,672
2910 DATA 25,0E,7E,A7,C8,06,20,FE,836
2920 DATA 20,C0,23,7E,10,F9,C9,2A,893
2930 DATA 8D,0D,C9,11,12,0E,7E,FE,784
2940 DATA 30,23,28,FA,2B,7E,A7,28,749
2950 DATA 0F,FE,20,28,0C,FE,61,38,760
2960 DATA 02,D6,20,1C,13,23,18,ED,581
2970 DATA 2B,AF,12,23,01,20,00,11,321
2980 DATA 25,0E,ED,B0,C9,47,FE,30,1038
2990 DATA D8,3E,46,B8,D8,78,FE,3A,1180
3000 DATA 38,05,FE,41,D8,D6,17,D6,1047
3010 DATA 20,A7,C9,E5,21,8D,0D,ED,1053
3020 DATA 6F,23,ED,6F,E1,C9,CD,27,1164
3030 DATA 09,06,08,7E,A7,CA,27,0A,567
3040 DATA FE,2E,CA,12,0A,11,17,0E,584
3050 DATA 7E,4F,A7,28,27,FE,20,28,777
3060 DATA 23,FE,2E,28,1F,FE,61,38,813
3070 DATA 02,D6,20,CD,D4,09,12,13,711
3080 DATA 23,10,E5,06,03,7E,FE,20,701
3090 DATA 28,5E,A7,28,5B,FE,2E,28,772
3100 DATA 29,C3,12,0A,3E,20,12,13,395
3110 DATA 10,FA,7E,FE,2E,28,1B,00,759
3120 DATA 06,03,18,44,4F,E5,D5,11,639
3130 DATA AA,0B,21,50,0A,7E,B9,CA,817
3140 DATA 06,0B,A7,23,20,F7,D1,E1,932
3150 DATA 79,C9,23,06,03,7E,A7,28,699
3160 DATA 21,FE,20,28,1D,FE,20,28,714
3170 DATA 1F,A7,28,1C,FE,61,38,02,675
3180 DATA D6,20,CD,D4,09,12,23,13,744
3190 DATA 7E,10,EA,FE,20,28,0F,A7,884
```

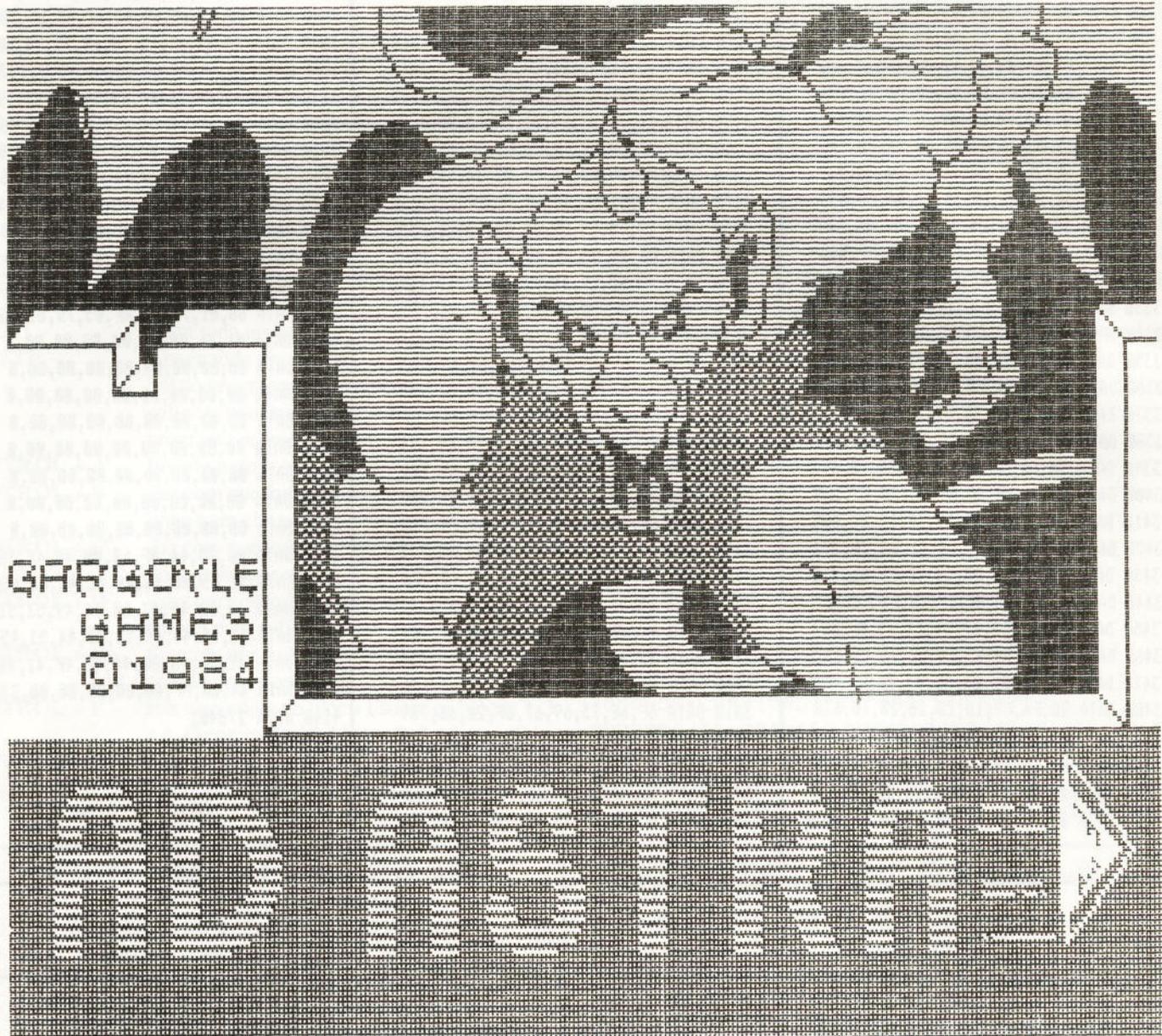
```
3200 DATA 28,0C,11,AA,0B,C3,06,0B,462
3210 DATA 3E,20,12,13,10,FA,11,25,451
3220 DATA 0E,01,20,00,ED,B0,C9,21,694
3230 DATA 05,0E,7E,11,AA,0B,FE,20,629
3240 DATA CA,06,0B,A7,CA,06,0B,11,622
3250 DATA 17,0E,01,08,00,ED,B0,23,494
3260 DATA 01,03,00,ED,B0,3E,1D,CD,713
3270 DATA 11,0B,11,05,0E,C3,24,0B,306
3280 DATA 2C,2E,2F,5C,7C,3F,2A,3B,517
3290 DATA 3A,5B,5D,2B,3D,22,00,CD,585
3300 DATA 18,0B,21,25,0E,22,8F,0D,309
3310 DATA 36,20,3E,3E,CD,11,0B,3E,505
3320 DATA 5F,CD,11,0B,CD,CD,0A,FE,1002
3330 DATA 0D,28,42,FE,1D,28,04,FE,700
3340 DATA 08,20,15,CD,DD,0A,A7,28,704
3350 DATA EB,2A,8F,0D,2B,22,8F,0D,666
3360 DATA 11,F3,0A,CD,24,0B,18,D7,761
3370 DATA 47,FE,20,38,D7,3E,A3,B8,1037
3380 DATA 78,38,D1,CD,DD,0A,FE,20,1107
3390 DATA 30,CA,3E,1D,CD,11,0B,78,694
3400 DATA CD,11,0B,2A,8F,0D,70,23,578
3410 DATA 22,8F,0D,18,B2,3E,1D,CD,688
3420 DATA 11,0B,3E,20,CD,11,0B,2A,397
3430 DATA 8F,0D,36,00,C9,1E,FF,0E,710
3440 DATA 06,CD,05,00,B7,28,F6,FE,939
3450 DATA 1B,CA,B3,07,C9,11,25,0E,684
3460 DATA 2A,8F,0D,A7,ED,52,7D,C9,1010
3470 DATA 1E,FF,0E,06,CD,05,00,B7,698
3480 DATA 28,F6,C9,1D,1D,20,20,1D,638
3490 DATA 1D,00,21,47,0B,11,74,F9,526
3500 DATA 01,34,00,ED,B0,C9,CD,18,896
3510 DATA 0B,CD,24,0B,C3,C9,07,3E,728
3520 DATA 0C,FD,21,BC,08,C3,74,F9,1054
3530 DATA D5,11,21,0B,CD,24,0B,D1,735
3540 DATA C9,0D,0A,00,EB,7E,EB,A7,987
3550 DATA C8,FD,21,BC,08,CD,74,F9,1252
3560 DATA 13,18,F1,7C,92,C0,7D,93,1018
3570 DATA C9,FD,21,E6,09,C3,74,F9,1286
3580 DATA FD,21,33,0A,C3,74,F9,F3,1150
3590 DATA ED,73,99,F9,31,F0,FA,08,1301
3600 DATA DB,A8,32,9B,F9,E6,FC,D3,1534
3610 DATA A8,08,CD,96,F9,08,3A,9B,1001
3620 DATA F9,D3,A8,08,ED,7B,99,F9,1398
3630 DATA C9,FD,E9,C9,00,00,00,00,888
3640 DATA 00,00,00,DB,A8,E6,F0,D3,1068
3650 DATA A8,DD,E9,41,72,71,75,69,1136
3660 DATA 76,6F,20,69,6C,65,67,61,775
3670 DATA 6C,00,44,61,64,6F,73,20,631
3680 DATA 69,6E,73,75,66,69,63,69,858
3690 DATA 65,6E,74,65,73,00,44,61,708
3700 DATA 64,6F,20,69,6C,65,67,61,757
3710 DATA 6C,00,4E,6F,6D,65,20,69,644
3720 DATA 6C,65,67,61,6C,00,53,65,701
3730 DATA 6D,20,61,72,71,75,69,76,805
3740 DATA 6F,73,00,44,69,72,65,74,730
3750 DATA 6F,72,69,6F,20,63,68,65,777
3760 DATA 69,6F,00,4D,65,6D,6F,72,728
3770 DATA 69,61,20,64,69,73,70,6F,777
3780 DATA 6E,69,76,65,6C,20,69,6E,789
3790 DATA 73,75,66,69,63,69,65,6E,854
3800 DATA 74,65,00,4E,61,6F,20,63,634
3810 DATA 6F,6E,73,69,67,6F,20,66,789
3820 DATA 65,63,68,61,72,20,6F,20,690
3830 DATA 61,72,71,75,69,76,6F,00,775
3840 DATA 45,72,72,6F,20,64,65,20,673
3850 DATA 6C,65,69,74,75,72,61,00,758
3860 DATA 45,72,72,6F,20,64,65,20,673
3870 DATA 67,72,61,76,61,63,61,6F,836
3880 DATA 00,43,6F,6D,61,6E,64,6F,705
3890 DATA 20,69,6E,65,78,69,73,74,804
3900 DATA 65,6E,74,65,00,41,72,71,720
3910 DATA 75,69,76,6F,20,69,6E,65,799
3920 DATA 78,69,73,74,65,6E,74,65,884
3930 DATA 00,20,20,00,53,49,53,00,303
3940 DATA 4D,53,58,20,44,65,62,75,664
3950 DATA 67,20,56,65,72,73,61,6F,759
3960 DATA 20,31,2E,30,0D,0A,43,50,345
3970 DATA 55,20,2D,20,30,31,2F,38,394
3980 DATA 39,00,D3,07,22,01,03,01,314
3990 DATA 66,01,73,01,AF,05,7D,06,530
4000 DATA 01,66,00,00,00,00,00,00,7
4010 DATA 00,00,00,00,00,00,00,00,0
4020 DATA 00,00,00,00,00,00,00,00,0
4030 DATA 00,00,00,00,00,00,00,00,0
4040 DATA 00,00,00,00,00,00,00,00,0
4050 DATA 00,00,00,00,00,00,00,00,0
4060 DATA 00,00,00,00,00,00,00,00,0
4070 DATA 00,00,00,00,00,00,00,00,0
4080 DATA 00,00,44,4F,53,00,4D,4F,386
4090 DATA 56,45,00,46,49,4C,4C,00,450
4100 DATA 45,58,45,43,00,44,49,53,517
4110 DATA 50,00,44,49,52,00,44,53,454
4120 DATA 41,56,45,00,44,4C,4F,41,508
4130 DATA 44,00,FF,00,00,00,00,00,323
4140 DATA 276403
```


CONVITE AOS NÃO SOCIOS DO MISC

Seja você um usuário profissional ou de lazer. convidamos a participar do melhor clube de MSX do Brasil. A proposta fundamental do MISC é dar apoio a nível de informação, serviço e produto.
Esta retaguarda é proporcionada por uma jovem equipe de especialistas em MSX. O Jornal do MSX é uma exclusividade dos associados. Para ingressar no MISC você paga uma taxa única de Ncz$ 6,50 até o fim do congelamento.
Pague com cheque nominal a EMBASS EDITORA LTDA., ou através de depósito no BRADESCO agência 0108 conta 141.184-5
Ao fazer sua inscrição informe-nos se quer receber sua coleção-brinde de jogos em K7 ou disco 5 1/4.

MISC - MSX INTERNATIONAL SERVICE CLUB
Rua Xavier de Toledo, 210 - cj. 23
01048 - São Paulo - SP - FONE: (011) 34-8391 e 36-3226

DUMP DE TELA GRÁFICA

JÚLIO VELLOSO

Para o usuário que não possui uma impressora colorida, desenvolvi uma rotina que imprime uma tela gráfica com diversas tonalidades de preto, dando um aspecto muito bonito a sua tela, que antes, com os "dump's" comuns, só aparecia com apenas uma cor e, em certos casos, quando o desenho exigia várias cores, ficava em parte cortado, perdendo muito na sua beleza.

A ROTINA

A rotina feita em assembler, ficando de B000h até BFE1H, imprime uma tela gráfica (modo 'SCREEN 2') em tamanho dobrado, utilizando 15 matrizes 2x2, substituindo as 15 cores do micro MSX por suas respectivas tonalidades de preto, como se fosse uma televisão preto e branco.

Como a impressora GRAFIX MTA possui 480 pontos para a impressão gráfica e a tela no modo gráfico possui 256 pontos na horizontal, a tela impressa ficaria com 512 pontos, cortando, portanto, 3.072 pontos de sua tela. Se a impressão fosse correta, os usuários desta impressora ficariam prejudicados. Então, eu optei pela impressão deitada, que são 192 linhas que impressas, por esta rotina, ficaria com 384 pontos podendo, portanto, ser efetuada.

MONTANDO A ROTINA

Para entrar com a rotina, digite o programa Basic da listagem 1 e execute-o com o comando RUN. O programa irá montar a rotina nos seus respectivos endereços e, em seguida, gravá-la em formato binário. Feito isto, o programa Basic não lhe será mais útil, podendo, portanto, ser apagado. Antes, é aconselhável testar a rotina.

UTILIZANDO A ROTINA

Para testar a rotina, entre com a listagem 2 e o programa montará uma tela com todas as cores do micro e em seguida efetuará a sua impressão.

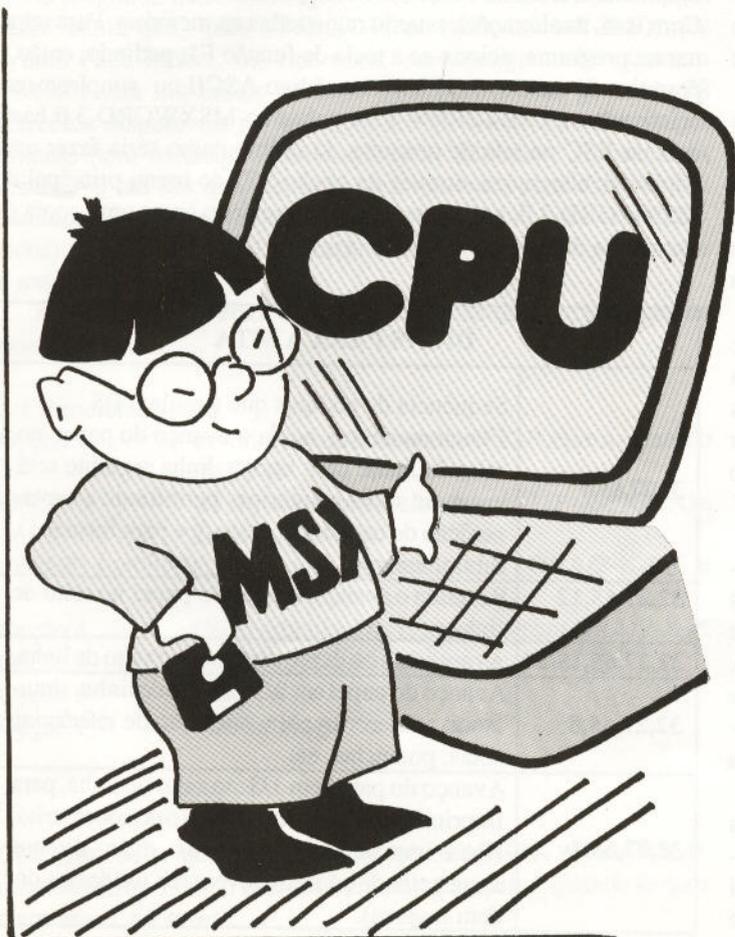
Para imprimir outras telas gráficas, como as geradas pelo GRAPHOS III, por exemplo, ou pelo comando BSAVE "NOME", em disco, entre com o seguinte programa em Basic:

```
10 SCREEN 2
20 BLOAD "NOME DA TELA",R (ou ,S - conforme o caso)
30 BLOAD"DUMP.EXE",R
```

E bons desenhos!

Júlio Velloso é estudante da UCP, na qual cursa o 4º período do Curso de Ciência da Computação. Programa em Basic, Assembler, Pascal e DBase III, há 4 anos, em equipamentos que utilizam o microprocessador Z80 e PC, principalmente o MSX.
É autor do programa utilitário de cópia BKP, comercializado inicialmente pela Ciberne e do Sistema Operacional BKP Disco, comercializado pela Paulisoft Informática.

```
10 '
20 '  Programa para impresao de tela
30 '
40 '      Por: Julio Velloso
50 '
60 FORI=1TO481:READA$:POKE &HBDFF+I,VAL
("&H"+A$):NEXTI
70 BSAVE"DUMP.EXE",&HBE00,&HBFE1
80 END
100 DATA DB,A8,47,E6,30,4F,CB,39,CB,39,
78,E6,F0,B1,D3,A8,3E,1B,CD,A5,0,3E,41,C
D,A5,0,3E,8,CD,A5,0,CD,61,BF,3A,EA,F3,2
1,CE,BF,E6,F,5F,16,0,19,7E,21,CE,BF
110 DATA 77,6,20,21,F8,0,C5,6,8,E5,C5,C
D,5B,BE,3A,E0,BF,3C,32,E0,BF,C1,E1,10,F
0,AF,32,E0,BF,11,8,0,A7,ED,52,C1,10,E0,
C3,B1,BF,6,C0,11,F8,0,CD,9C,BE,CD
120 DATA BE,BE,E5,D5,C5,CD,AD,BE,C1,D1,
E1,7D,E6,7,FE,7,20,1,19,23,10,E6,C9,3E,
1B,CD,A5,0,3E,4B,CD,A5,0,7B,CD,A5,0,7A,
CD,A5,0,C9,C5,E,0,6,8,CB,17,CB
130 DATA 19,10,FA,79,C1,C9,CD,A2,BE,DB,
98,C9,7D,F3,D3,99,7C,E6,3F,D3,99,FB,C9,
F5,3A,E0,BF,A7,CA,CA,BF,47,F1,17,10,FD,
CB,7F,20,17,EB,21,0,20,19,CD,9C,BE,E6
140 DATA F,21,CE,BF,5F,16,0,19,7E,4F,CD
,E5,BE,C9,EB,21,0,20,19,CD,9C,BE,E6,F0,
1F,1F,1F,1F,18,E3,AF,C5,CB,59,20,4,CB,B
F,18,2,CB,FF,CB,51,20,4,CB,B7,18,2,CB
150 DATA F7,4F,3A,DE,BF,A7,CA,C2,BF,47,
79,1F,1F,10,FC,4F,FD,7E,0,B1,FD,77,0,FD
,23,AF,C1,CB,49,20,4,CB,BF,18,2,CB,FF,C
B,41,20,4,CB,B7,18,2,CB,F7,4F,3A,DE
160 DATA BF,A7,CA,C6,BF,47,79,1F,1F,10,
FC,4F,FD,7E,0,B1,FD,77,0,FD,23,3A,DF,BF
,3C,32,DF,BF,FE,C0,28,1,C9,AF,32,DF,BF,
FD,21,0,40,3A,DE,BF,3C,32,DE,BF,FE,4
170 DATA 28,15,C9,E5,D5,C5,21,0,40,36,0
,11,1,40,1,0,30,ED,B0,C1,D1,E1,C9,AF,32
,DE,BF,D5,C5,1,80,1,FD,21,0,40,11,80,1,
CD,7B,BE,FD,7E,0,CD,A5,0,CD,A7
180 DATA BF,FD,23,B,78,B1,20,F0,3E,A,CD
,A5,0,C1,D1,FD,21,0,40,CD,61,BF,C9,CD,B
7,0,D0,6,B,FD,E1,10,FC,DB,A8,E6,F0,D3,A
8,3E,A,CD,A5,0,3E,D,CD,A5,0,C9
190 DATA 79,C3,9,BF,79,C3,37,BF,F1,C3,B
A,BE,7,F,A,9,7,5,D,8,C,6,4,2,B,3,1,0,0,
0,0,0
```

```
10 '
20 '  PROGRAMA TESTE
30 '
40 COLOR1,15,15:SCREEN 2
50 X=20:Y=18
60 FORC=1TO15
70 LINE(X,Y)-(X+36,Y+24),C,B:PAINT(X+10
,Y+10),C
80 X=X+80:IFX>180THENY=Y+32:X=20
90 NEXT C
100 BLOAD"DUMP.EXE",R
```

PACOTE Nº 1
10 jogos — apenas 11,00 (disco incluído)

PACOTE Nº 4
1 disco cheio — 11,00
2 disco cheios — 20,00

PACOTE Nº 2
10 jogos — apenas 11,50 (fita incluída)

PACOTE Nº 5
Aplicativo + copiador — 6,00 (disco incluído)

PACOTE Nº 3
MSX TOOLS I e II 5,00 sucesso de vendas

PACOTE Nº 6
10 Aplicativos + disco — apenas 28,00

LANÇAMENTOS DO MÊS
MSX BOOK II - EDTRONIC - SCED
SUPER ASM (CART C./LED E'RESET)

A cada 5 programas escolha 1 grátis

Para outros estados, os pedidos deverão ser feitos através de cheque nominal e cruzado a MARCO ANTONIO TROVÃO VAZ, Rua Carvalho Alvim 278/501, Rio de Janeiro, RJ, CEP 20510. Tel: (021) 268-6360

REDEFININDO O MSXWORD 3.0

GUSTAVO F. BAYER

Uma das grandes vantagens dos editores de texto baseados no clássico TASWORD inglês é a possibilidade de total redefinição dos caracteres impressos na tela, assim como dos respectivos códigos gravados e remetidos à impressora. Sua mais recente versão comercializada no Brasil, o MSXWORD 3.0, apresenta rotinas que facilitam a redefinição dos códigos emitidos por cada caracter, mas a alteração dos caracteres na tela continua sendo pouco aproveitada pelos usuários, já que o acesso à sua tabela binária é trabalhoso.

O programa aqui apresentado busca facilitar o manuseio conjunto de todas as tabelas referentes a cada caracter. Ele é compatível apenas com o MSXWORD 3.0, já que essa versão, para facilitar a carga do programa, alterou os endereços de suas rotinas em assembler.

Inicialmente, o programa pede que seja informado o caracter ou o código ASCII a ser pesquisado. A seguir, ele apresenta, de cima para baixo, os 8 bytes que definem o caracter na tela de 64 colunas, os 8 bytes que definem o caracter na janela de 32 colunas, e, ao final, os códigos gravados e emitidos à impressora. Nessa tela de pesquisa, os bytes são apresentados de forma simplificada, eliminando-se os"zeros", para facilitar o reconhecimento do desenho do caracter. Os códigos para gravação e impressão são apresentados em sua notação decimal.

A pesquisa pode continuar através das teclas de direção: a tecla para cima avança ao topo da tabela (retornando em um caracter por toque); a tecla para baixo desce na tabela (avançando um caracter por toque); e as teclas esquerda ou direita solicitam a informação de outro caracter ou código ASCII a ser pesquisado. Os caracteres ASCII 208 a 223 não são apresentados por não estarem disponíveis para redefinições.

Ao pressionar-se a tecla INS, surge a tela de trabalho, onde os bytes que definem o caracter na tela são apresentados em sua notação binária completa, onde os algarismos "1" representam os pontos que serão impressos na tela. Basta, então, acrescentar ou retirar tais algarismos para modificar-se o caracter no MSX WORD 3.0.

Alguns cuidados, entretanto, devem ser tomados. Os caracteres da tela de 64 colunas ocupam apenas as quatro colunas à direita dos seus bytes. Para manter uma separação entre os caracteres, apenas as três últimas à direita devem ser utilizadas. Na tela de 32 colunas, os caracteres ocupam todo o byte, mas apenas os seis à esquerda devem ser utilizados para manter-se o afastamento. Cuidado com o caracter ASCII 197, que define o cursor da janela de 32 colunas.

Os códigos para gravação e impressão são alterados com a simples troca do respectivo número em notação decimal. É importante observar aqui que cada caracter normal (sem o GRAPH acionado) emite uma sequência de três códigos. Essa quantidade de códigos é suficiente para acionar-se a maior parte dos comandos de uma impressora.

Isso permite utilizar caracteres praticamente nunca usados em textos, especialmente os do teclado CODE (Hotbit) ou shift/RGRA (Expert), reservando-se os caracteres obtidos no MSXWORD 3.0 com GRAPH acionado, e que emitem quatro códigos, para os comandos mais extensos.

Apenas uma observação: deve-se procurar incluir na sequência dos códigos de controle da impressora o código 32 (espaço). Isso permitirá incluir o respectivo caracter em qualquer espaço da linha, sem perder o alinhamento à direita na impressão (principalmente quando se trabalha com todas as 64 colunas de texto).

Combinando-se esses recursos de redefinição, torna-se possível criar caracteres de controle simbolizando na tela sua função, facilitando assim o controle visual do texto na tela.

Uma vez efetuadas as modificações desejadas, basta colocar o cursor na última linha alterada da tela de trabalho e acionar repetidamente a tecla de função F2 até aparecer a mensagem de erro. Com isso, as alterações estarão registradas na memória. Para retornar ao programa, aciona-se a tecla de função F1, pedindo, então, a pesquisa de um novo caracter ou código ASCII ou simplesmente acionando-se <RETURN>. Para voltar ao MSXWORD 3.0 basta acionar ESC na tela de pesquisa. O último passo seria fazer uma cópia das alterações, através da opção "C" do menu principal do MSXWORD 3.0, sem esquecer-se de ter feito antes uma cópia de segurança do MSXWORD 3.0 original.

DICAS PARA A MTA	
32,27,65,0	Sequência de códigos que simula o BS. Funcionamento: anula o avanço do papel no salto de linha; com isso, a linha seguinte será impressa sobre a anterior, permitindo a superposição de caracteres (p.ex./e = para formar ‡), sublinhamento e duplo (triplo ...) toque.
32,27,65,12	Restaura o avanço normal do papel no salto de linha.
32,27,65,18/24	Avanço 1,5 ou duplo do papel no salto de linha.
32,27,65,8	Avanço do papel em 2/3 no salto de linha, simulando sobrescrito para anotação de referências, notas, potências, etc.
32,27,65,4	Avanço do papel em 1/3 no salto de linha, para imprimir a linha normal abaixo do sobrescrito. (Para imprimir subscrito, as duas últimas sequências de códigos devem ser usadas na ordem inversa).

Especialmente interessante é a simulação do negrito.

Para tanto necessitamos inicialmente a simulação do BS na linha principal. No início da linha seguinte, precisamos da sequência:

27, 75, 1, 0,

que prepara a impressora para a impressão de uma única coluna de pontos no modo gráfico.

A seguir, precisamos da sequência:

0, 32, 32

que provoca o avanço de uma única coluna de pontos sem nada imprimir, saltando a seguir dois espaços. Com isso, basta repetir a palavra a ser impressa em negrito exatamente abaixo de sua ocorrência na linha anterior.

Para imprimir negrito no início da linha, a sequência:

0, 32, 32

deve ser substituída por um caracter nulo, que emite apenas o código 0.

Finalmente, não pode faltar o código **12**, que provoca o avanço de página, permitindo assim a paginação da impressão na tela, antes da impressão (principalmente se for usada a simulação do BS).

```
10 '******************************
20 '* Redefinidor de caracteres *
30 '*  e códigos de impressão   *
40 '*    para o MSXWORD 3.0     *
50 '*       da Cibertron        *
60 '******************************
70 WIDTH40:KEY1,CHR$(30)+CHR$(21)+"run7
0"+CHR$(13):KEY2,CHR$(13)+"poke"+CHR$(3
0)+CHR$(30)+CHR$(30):KEY3,"00":KEY4,"00
0":KEY5,"0000":KEYOFF
80 K$=" ":LINEINPUT"QUAL CARACTER OU Có
DIGO? ";L$
90 IFLEN(L$)=1THENC=ASC(L$)ELSEC=VAL(L$
)
100 IFC<32THENC=32
110 IFC=>208ANDC<=223THENIFASC(K$)=30TH
ENC=207ELSEC=224
120 IFC=>255THENC=255
130 CC=C:L$=CHR$(C)
140 CLS:PRINT" CHR$ ";L$;SPC(7);"765432
10";SPC(4);"ASC ";C:B=44800!+8*C:A=B:FO
RR=BTOB+7:GOSUB290:NEXTR
150 B=42752!+8*C:A=B:FORR=BTOB+7:GOSUB2
90:NEXTR
160 IFC<=191THENC=&H9BBF+3*(C-32)ELSEIF
C=>208THENC=&H9D9F+3*(C-224)ELSEC=&HA4C
9+(C-64)*4
170 PRINT"POKE";C;",";PEEK(C)
180 PRINT"POKE";C+1;",";PEEK(C+1)
190 PRINT"POKE";C+2;",";PEEK(C+2)
200 IFC=>-22839ANDC<=-22778THENPRINT"PO
KE";C+3;",";PEEK(C+3)
210 IFASC(K$)=18THENPRINT"Ok. Feitas as
 modificações, registre-as   usando <F
2> a partir da última linha":PRINT"  a
lterada.  <F1> reinicia o programa.";CH
R$(30);CHR$(30);CHR$(30);:END
220 PRINT"CURSOR p/pesquisar       INS p
/modificar":PRINT"         ESC retorna a
o MSXWORD"
230 K$=INKEY$:IFK$=""THEN230
240 IFASC(K$)=31THENC=CC+1:CLS:GOTO100
250 IFASC(K$)=30THENC=CC-1:CLS:GOTO100
260 IFASC(K$)=18THENC=CC:LOCATE0,1:GOTO
140
270 IFASC(K$)=27THENKEY1,CHR$(&HD8):KEY
2,CHR$(&HDA):KEY3,CHR$(12):LOAD"MSXWORD
3.0",R
280 PRINTCHR$(30):GOTO80
290 A$=BIN$(PEEK(A))
300 IFLEN(A$)<8THENFORF=1TO(8-LEN(A$)):
A$="0"+A$:NEXTF
310 FORF=8TO1STEP-1:B$(F)=MID$(A$,F,1):
IFB$(F)="0"THENB$(F)=" "
320 NEXTF
330 C$="":FORF=8TO1STEP-1:C$=B$(F)+C$:N
EXTF
340 PRINT"POKE";A;",&b";:IFASC(K$)=18TH
ENPRINTA$ELSEPRINTC$
350 A=A+1:RETURN
```


TOYGAMES INFORMATICA

MSX

A TOYGAMES INFORMATICA DISPÕE DOS MELHORES JOGOS PARA O SEU MSX, OFERECENDO QUALIDADE PROFISSIONAL, NOVIDADES INTERNACIONAIS E GARANTIA DE SEUS SERVIÇOS.

SOLICITE NOSSO CATÁLOGO GRÁTIS

FONE - (011)289-5630 - CAIXA POSTAL: 30961 - CEP: 01051 - SP/SP

PACOTE GRÁFICO PARA PASCAL

PAULO MARQUES FIGUEIRA

Apresentamos um pequeno pacote de procedures para o TURBO PASCAL com o qual os usuários desta linguagem poderão usar certos comandos que só encontramos no Basic do MSX. São comandos para a produção de sons (Play e Sound) e para recursos gráficos (Line, Circle, etc.), que, normalmente, não estão disponíveis em linguagens compiladas vindas de sistemas como o CP/M.

O recurso mais usado do Pascal foi o INLINE, que nos permite colocar rotinas em linguagem de máquina dentro dos procedures e, assim, usar muitas rotinas da própria ROM do MSX.

É importante ressaltar que os comandos gráficos criados utilizam o interpretador Basic da máquina. Logo, o complemento dos comandos e a velocidade de processamento é igual à do Basic. Cuidado para não cometer erros nos comandos, pois o interpretador apresentará a mensagem de erro e o controle do micro passa para o Basic. Portanto, sempre salve os programas antes de rodá-los, pela opção R do menu principal do Pascal.

No processo usado é criado uma falsa linha de programa Basic e mandamos que o interpretador a cumpra. Devido a isto, esta falsa linha tem que ficar em uma área da memória acima da ROM, que também deverá estar totalmente ativada. Assim, foi definido uma variável para o complemento dos comandos, alocada no endereço $A000 e uma rotina em linguagem de máquina no endereço $C000, que fez a mudança dos slots, ativando as páginas da ROM.

Em outros comandos, como os de mudança de screen, são simplesmente chamadas as rotinas do BIOS. Note que os programas compilados em Pascal ficam no endereço $100 e, portanto, estão com as páginas de ROM desligadas. Mesmo assim, dispomos de rotinas, como a do endereço $30, que faz chamadas entre slots. Desta forma, o RST $30 foi muito utilizado.

Na listagem 1 temos as procedures criadas e, nas demais listagens, temos exemplos de uso. Observe que a diretiva de compilação (*$I*), que faz a inclusão de fontes no programa, incluirá todos os procedures no seu programa, quer elas sejam usadas ou não. Assim, se preferir, grave as procedures no seu disco separadamente e só inclua as que for usar no seu programa.

No caso do comando LINE, a única diferença é que o traço que liga as coordenadas dos vértices deve ser o token do sinal de menos. Observe os programas exemplos.

Espero que, com isso, os usuários no MSX possam tirar melhor proveito desta linguagem, "envenenando" seus programas com os recursos do MSX.

Até à próxima!

Paulo Marques Figueira programa em Basic, Cobol, Pascal, dBase e Assembly, em micros da linha MSX. É o autor do programa Edtronic, comercializado pela Paulisoft, Softhouse para a qual desenvolve vários projetos.

```
VAR SLOT: STRING[19] ABSOLUTE $BFFF;
    COMP: STRING[250] ABSOLUTE $9FFF;
PROCEDURE SCREEN0;
BEGIN
    INLINE ($F7/$00/$6C/$00);
END;

PROCEDURE SCREEN1;
BEGIN
    INLINE ($F7/$00/$6F/$00);
END;

PROCEDURE SCREEN2;
BEGIN
    INLINE ( $F7/$00/$72/$00);
END;

PROCEDURE SCREEN3;
BEGIN
    INLINE($F7/$00/$75/$00);
END;

PROCEDURE PLAY;
BEGIN
    COMP:=COMP+CHR(0);
    INLINE($21/$00/$A0/$DD/$21/
            $E5/$73/$CD/$00/$C0);
END;

PROCEDURE CIRCLE;
BEGIN
    COMP:=COMP+CHR(0);
    INLINE($21/$00/$A0/$DD/$21/
            $11/$5B/$CD/$00/$C0);
END;

PROCEDURE LINE;
BEGIN
    COMP:=COMP+CHR(0);
    INLINE($21/$00/$A0/$DD/$21/
            $0E/$4B/$CD/$00/$C0);
END;

PROCEDURE DRAW:
BEGIN
    COMP:=COMP+CHR(0);
    INLINE($21/$00/$A0/$DD/$21/
            $6E/$5D/$CD/$00/$C0);
END;

PROCEDURE PAINT;
BEGIN
```

```
    COMP:=COMP+CHR(0);
    INLINE($21/$00/$A0/$DD/$21/
           $C5/$59/$CD/$00/$C0);
END;

PROCEDURE PSET;
BEGIN
    COMP:=COMP+CHR(0);
    INLINE($21/$00/$A0/$DD/$21/
           $EA/$57/$CD/$00/$C0);
END;

PROCEDURE VDPON;
BEGIN
    INLINE($F7/$00/$44/$00);
END;

PROCEDURE VDPOFF;
BEGIN
   INLINE ($F7/$00/$41/$00);
END;

PROCEDURE PRINTGRP;
BEGIN
    COMP:=COMP+CHR(0);
    INLINE($21/$00/$A0/$7E/$FE/
           $00/$CB/$F7/$00/$8D/
           $00/$23/$18/$F5);
END;

PROCEDURE COLOR(CORI,CORF,CORB:BYTE);
BEGIN
    MEM[$F3E9]:=CORI;
    MEM[$F3EA]:=CORF;
    MEM[$F3EB]:=CORB;
    INLINE($F7/$00/$62/$00);
END;

PROCEDURE VPOKE(ENDE:INTEGER;DADO:BYTE);
VAR ENDE1: INTEGER ABSOLUTE $A000;
   DADO1: BYTE ABSOLUTE $B002;
BEGIN
  ENDE1:=ENDE; DADO1:=DADO;
  INLINE($3A/$02/$B0/$2A/$00/
         $A0/$F7/$00/$4D/$00);
END;

FUNCTION VPEEK(ENDE:INTEGER) : INTEGER;
VAR ENDE1: INTEGER ABSOLUTE $A000;
   X: BYTE ABSOLUTE $A002;
BEGIN
    ENDE1:=ENDE;
    INLINE($2A/$00/$A0/
           $F7/$00/$4A/$00/
           $32/$02/$A0);
    VPEEK:=X;
END;

PROCEDURE SOUND(REGISTRO,DADO:BYTE);
BEGIN
    PORT[160]:=REGISTRO;
    PORT[161]:=DADO;
END;

PROCEDURE WIDTH(X:BYTE);
BEGIN
    MEM[$F3B0]:=X;
END;

PROCEDURE CURSOR(X:BYTE);
BEGIN
    MEM[$FCA9]:=X;
END;

PROCEDURE FILES;
(*COMP:='"*.*"';FILES; *)
BEGIN
    COMP:=COMP+CHR(0);
    INLINE($21/$00/$A0/$DD/$21/
           $2F/$6C/$CD/$00/$C0);
END;

(*INICIO DO PROGRAMA*)
BEGIN
    SLOT:=CHR($DB)+CHR($A8)+CHR($F5)+CHR($E6)
         +CHR($F0)+CHR($D3)+CHR($A8)+CHR($FD)
         +CHR($21)+CHR($00)+CHR($00)+CHR($CD)
         +CHR($1C)+CHR($00)+CHR($F1)+CHR($D3)
         +CHR($A8)+CHR($C9);
END;
```

```
EXEMPLO 1
PROGRAMA LINHA;
VAR A: INTEGER;
   B: STRING[3];
         {$I MSXPACOT.PAG}
  COLOR(15,1,1);
  SCREEN2;
  COMP:='(10,10)'+CHR($F2)+'(250,180),7,B'; LINE;
  A:=5;
  REPEAT
  STR(A,B); COMP:='(128,80),'+B+',10'; CIRCLE;
  A:=A+5;
  UNTIL A>60;
  COMP:='(85,77)';PSET;
  COMP:='REVISTA CPU'; PRINTGRP;
  COMP:='(86,77)';PSET;
  COMP:='REVISTA CPU'; PRINTGRP;
  COMP:='"V15T255L3207CDCD"'; PLAY;
  READ(A);
  SCREEN0;
END.
```

```
EXEMPLO 2
VAR A: BYTE;
   (*$I MSXPACOT.PAS *)

  COMP:='"*.*"'; FILES;
  FOR A:=1 TO 40 DO BEGIN
     WIDTH(A);
     GOTOXY(1,1); DELLINE;
  END;
  FOR A:=1 TO 24 DO BEGIN
     GOTOXY(1,1); INSLINE;
  END;
  END.
```

```
EXEMPLO 3
  VAR W: BYTE;
  I:INTEGER;
   (*$I MSXPACOT.PAS*)
  SCREEN1;
  WRITELN('ESTOU FAZENDO... ESPERE...');
  FOR I:=264 TO 976 DO BEGIN
  W:=VPEEK(I);
  VPOKE(I,W OR (TRUNC(W/2)+TRUNC(W/4)));
  END;
  COMP:='"*.*"'; FILES;
  READ(W);
  SCREEN0;
  END.
```

RALLY PARIS-DAKAR

Para voce que gosta de fortes emocoes, um jogo que certamente vai tira-lo do serio! **PARIS-DAKAR** e' um jogo de simulacao automobilistica onde sera' necessario que o piloto demonstre tuda a sua pericia em "cross e enduro".
Em fita ou disco - NCz$ 7,00!

REDATOR ELETRONICO PARA MSX

O melhor **PROCESSADOR DE TEXTOS** ja' criado para o **MSX**! Em **CARTUCHO** original da GRADIENTE por apenas NCz$ 18,00 !!

MSX HARDCOPY

Um programa indispensavel para quem possui uma impressora e quer tirar o maximo proveito da mesma com diversos tipos e tamanhos de graficos! Imprime ainda em cor inversa, em sentidos e tabulacoes diferentes!
Em fita ou disco - NCz$ 9,50!

MSX-SAM

O programa que **faz o seu MSX falar**! Uma criacao da NEMESIS INFORMATICA!
Em fita ou disco - NCz$ 9,50!

MSX CHART

O melhor pacote de edicao de **GRAFICOS COMERCIAIS** e **ESTATISTICOS** disponivel ate' entao para a linha **MSX**! Monta graficos **SETORIAIS**, de **AREA**, **LINEARES** e de **BARRA**; ou ainda **MIXTOS**. **IMPRIME** o resultado em qualquer impressora grafica e pode enviar telas para o **MSX PAGE MAKER 1.3**, ou editores graficos como o **YAMAHA ARTIST** ou **NEMESIS GRAF-MASTER**, **SONY GRAPHIC MASTER** ou **CARTOON**!

Um programa indispensavel para aqueles que utilizam o seu microcomputador em atividades mais profissionais.
Apenas em disco - NCz$ 14,00!

MSX PORTFOLIO

Um programa que reune num mesmo disco uma **AGENDA ELETRONICA**, uma **LISTA TELEFONICA**, um **CALENDARIO PERPETUO** e uma **CALCULADORA ELETRONICA**; formando assim um sistema de "DESK-TOP" eletronico para facilitar o seu dia-a-dia!

Acompanha o programa o ano de 1989 com seus feriados e datas comemorativas; pre-programados num disquete adicional gratuito! Aproveite!
Apenas em disco - NCz$ 14,00!

NOVO CATALOGO NEMESIS

REMETEMOS PARA TODO O BRASIL!

NEMESIS SPECIAL DISK PACK

A **NEMESIS** reuniu num disco dois jogos de peso: **THE FINAL COUNTDOWN** e **PLAY HOUSE STRIP POKER** em **VERSOES ORIGINAIS.**
Com a qualidade e exclusividade de costume da **NEMESIS**! Em disco por apenas NCz$ 8,00

NEMESIS SPECIAL GAME PACK 5

Um pacote especial reunindo os sensacionais jogos: **TAWARA, SNAKE IT, THE POWER OF DARKNESS e BOARDY GAMES II.** Nao perca estes lancamentos! Em fita ou disco - NCz$ 6,00

OS BEST-SELLERS DE 1988

MSX HELLO 1.0

O Super Sistema Operacional com diversos recursos e implementacoes ! Em disco por NCz$ 18,60

MSX PAGE MAKER 1.3

O programa que usamos para compor esta pagina! Uma revolucao para o **MSX**! Em disco por NCz$ 18,60

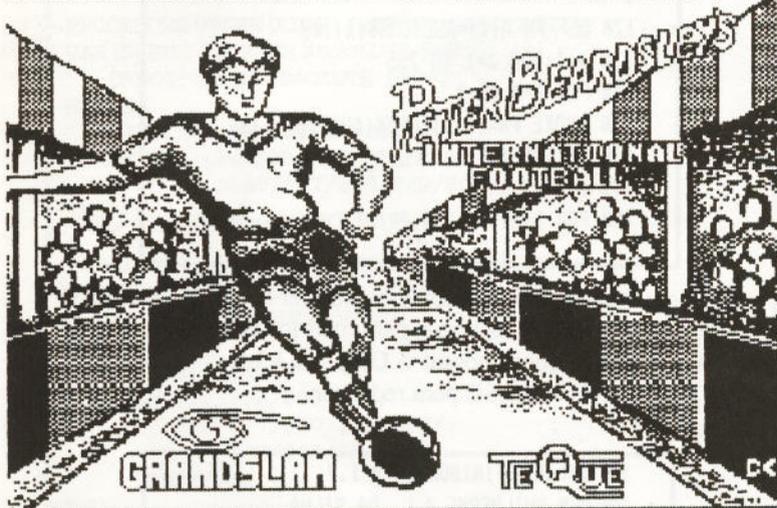
MSX PAGE MAKER KIT

3 DISCOS com **FIGURAS** e **LETRAS VARIADAS**: NCz$ 31,40

NOVIDADES PARA O SEU MSX2

Com a mesma tradicao em novidades e qualidade ja' conhecidas pelos usuarios de **MSX1**, a **NEMESIS** apresenta tambem as ultimas sensacoes do momento para o seu MSX de 2a geracao! Peca o nosso catalogo gratis!

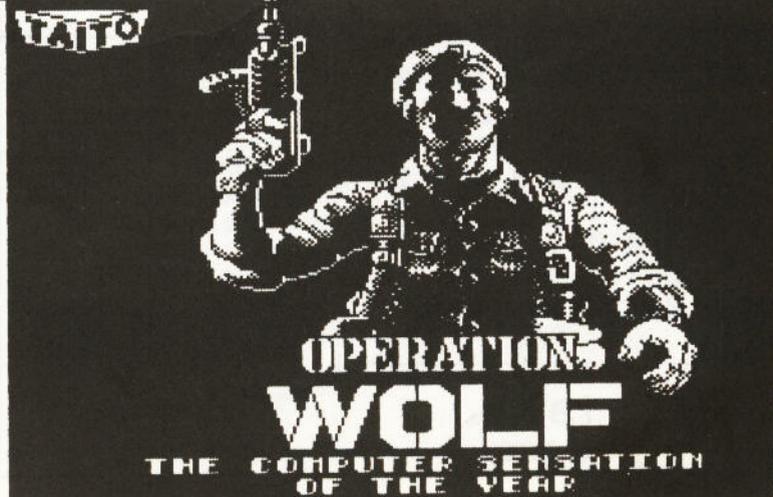
OPERATION WOLF EM DISCO

Um dos mais fantasticos jogos ja' criados para a linha **MSX**. Sua missao e' resgatar os prisioneiros de um campo de de concentracao nazista.
Agora em disco por NCz$ 7,00!

RESASAM A INTERFACE COMPLETA

Um **cartucho** com "**BOTAO DE RESET FISICO**", desarmando todos os programas; **ASSEMBLER E DISASSEMBLER; RECUPERADOR DE PROGRAMAS PERDIDOS, DE DISCOS AVARIADOS; ZAPPER; HARD-COPY DE TELAS GRAFICAS; ETC...**
Em **cartucho** por NCz$ 38,00!

NEMESIS INFORMATICA LTDA

Envie VALE POSTAL ou CHEQUE NOMINAL a NEMESIS INFORMATICA

CAIXA POSTAL 4.583 CEP 20.001 RIO DE JANEIRO - RJ.

ALFABETOS DO GRAPHOS III: APLICAÇÕES PRÁTICAS

GUILHERME A. L. STLLK

O programa editor gráfico Graphos III, versões 1.1 e 1.2, é um dos mais fáceis de se utilizar e um dos mais completos para a linha MSX.

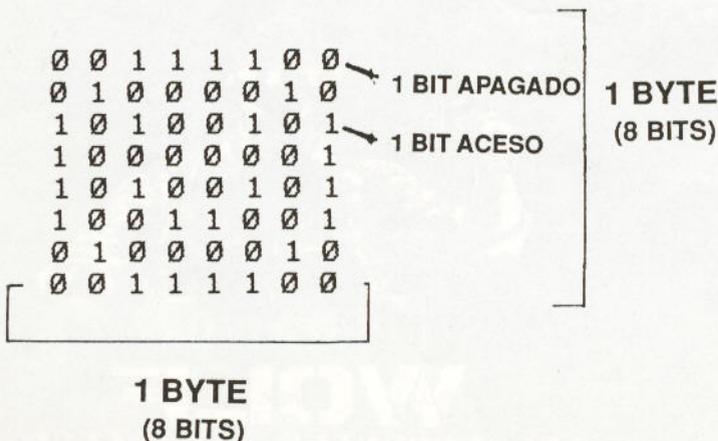
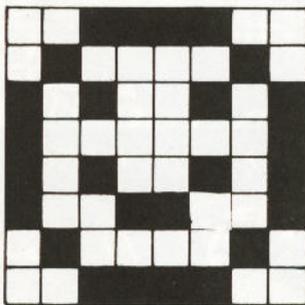
Um dos seus grandes recursos é o de poder editar alfabetos com uma simplicidade incrível. Quem tem, sabe.

A edição de alfabetos permite que o seu uso não se restrinja só ao programa fonte (no uso de edição de telas), mas também permite sua utilização no BASIC e Assembly Z80 com diversas aplicações, por exemplo: redefinição do alfabeto do MSX, impressão, utilização em outros programas e mil coisas que sua fértil imaginação mandar.

ORGANIZAÇÃO

Quando o Graphos III edita alfabetos, ele grava os bytes dos caracteres na memória a partir dos endereços 9000H (versão 1.1) e 9200H (versão 1.2). Os bytes são gravados assim: cada 8 bytes um caracter, cada byte corresponde a uma linha gráfica deste (são 8 linhas), que, por sua vez, é formada de 8 bits e cada bit corresponde a um ponto que pode estar aceso (1) ou apagado (0). Veja figura1. OBS: O sistema hexadecimal é representado por um número seguido de H, por exemplo, 9000H.

Um arquivo de alfabeto tem aproximadamente 200H de extensão e pode ser deslocado por um comando BLOAD"xxxx", deslocamento. O MSX faz esta operação adicionando ao endereço inicial o número após o BLOAD.

Figura 1

APLICAÇÕES

1. Redefinição

A redefinição do set de caracteres do MSX necessita de uma pequena rotina no BASIC, pois a redefinição não é feita por BLOAD "xxxx", R, como muitos pensam.

O programa utiliza a SCREEN 1 por ter um padrão de 8x8 pontos e dá VPOKEs na tabela de caracteres com os bytes do alfabeto.

Quem tiver a versão 1.1 do Graphos, tem que mudar a linha 150 de 9200H para 9000H.

Figura 2

```
10 REM REDEFINIDOR
20 REM
30 REM PARA A LINHA MSX
40 REM
50 REM GUILHERME A.L. DA SILVA
60 REM GUARARAPES - SP  13/9/88
70 REM
80 KEY 1,"SCREEN1"+CHR$(13)
90 SCREEN1
100 PRINT:PRINT:PRINT:FORI=1TO31:PRINTC
HR$(1);CHR$(65+V);:V=V+1:NEXTI
110 FOR I=32 TO 255:PRINT CHR$(I);:NEXT
I
120 DEF FN A(X)=ASC(CHR$(X))*8
130 X=1:FOR J=1 TO 255
140 FOR I=0 TO 7
150 VPOKE FNA(X)+I,PEEK(&H9200+I+J*8)
160 NEXT I:X=X+1
170 NEXT J
180 FOR I=32 TO 255:PRINT CHR$(I);:NEXT
 I
```

Como o Basic é muito lento, isso pode ser feito em LM Z80. O programa da figura 2.1 faz isso e, para rodá-lo, dê RUN e carregue um alfabeto. Para quem possui o Graphos 1.1, comande POKE &hE020,&h90. Aperte ESC para redefinir.

Figura 2.1

```
10 REM REDEFINIDOR LM V 1.2
20 REM GUILHERME A.L. DA SILVA
30 REM PARA MSX
40 REM
50 FOR I=&HE000 TO &HE040:READ A$:POKE
I,VAL("&H"+A$):NEXT:DEFUSR=&HE000:A=USR
(0)
60 REM
70 REM DADOS
100 DATA 21,0C,E0,22,CD,FD,3E,CD,32,CC,
FD,C9,FE,3A,C0,CD,6F,00,CD,7B,00,CD,53,
00,01
110 DATA 00,08,11,00,00,21,00,92,CD,5C,
00,06,0F,21,31,E0,7E,CD,A2,00,21,10,F9,
C9,52,45,44,45,46,49,4E,49,44,4F,0A,0D,
4F,6B,0A,0D
```

2. Impressão

A impressão de caracteres especiais ou redefinidos só pode ser feita no modo gráfico de sua impressora (ver artigo sobre impressão em modo gráfico em CPU nº 1). O uso dele não é muito fácil porque a impressora trabalha com os bytes na vertical, devido às agulhas da cabeça de impressão e o vídeo do MSX na horizontal. Para corrigir isso, uma rotina tem que "tombar" o padrão de bits dos caracteres (figuras 3 e 4).

Na rotina da figura 5 é feito exatamente isso, com o auxílio de uma pequena rotina em linguagem de máquina (figura 6) para agilizar o processo.

Uma 2ª rotina é utilizada para a impressão, que é composta por uma em Basic (figura 7) e uma outra em Linguagem de Máquina (figura 8).

Figura 3

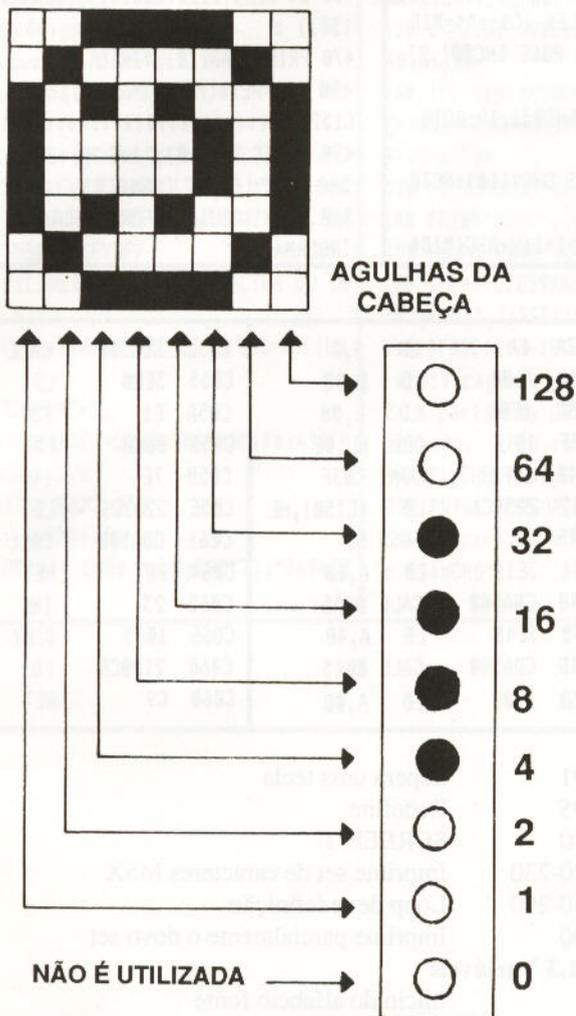



Figura 4

BYTES "TOMBADOS"

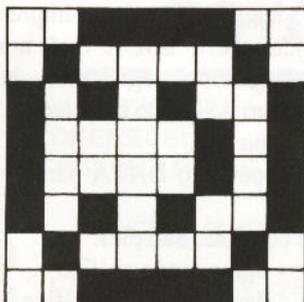



Figura 5

```
10 INVERSOR DE ALFABETOS
20 REM PARA O GRAPHOS III V 1.2
30 REM
40 REM PARA A LINHA MSX
50 REM
60 REM GUILHERME A. L. DA SILVA
70 REM GUARARAPES - 08/09/88
80 REM
81 DATA 11,F8,97,06,08,C5,21,F8,C9,06,0
8,1A,17,CB,16,23,10,FA,C1,13,10,EF,C9
82 FOR I=&HC000 TO &HC000+22:READ X$ :
POKE I,VAL ("&H"+X$) : NEXT
90 CLS
95 REM - INVERTE
100 FOR I=0 TO 2047 STEP 8
110 DEFUSR=&HC000
120 A=USR(0)
130 H=&H9200+I:H$=HEX$(H):LOCATE 10,10:
PRINT"ENDEREÇO FONTE :";H$
140 POKE &HC001,VAL("&H"+RIGHT$(H$,2))
150 POKE &HC002,VAL ("&H"+LEFT$(H$,2))
160 G=&HC800+I:G$=HEX$(G):LOCATE 10,12:
PRINT"ENDERECO DESTINO: ";G$
170 POKE &HC007,VAL ("&H"+RIGHT$(G$,2))
180 POKE &HC008,VAL("&H"+LEFT$(G$,2))
190 NEXT
191 LOCATE 0,21:PRINT"PRESSIONE QUALQUE
R TECLA PARA CONTINUAR";:C$=INPUT$(1)
195 REM REDEFINE
200 KEY 1,"SCREEN1"+CHR$(13)
210 SCREEN 1
220 CLS:LOCATE10,0:PRINT"TESTE":PRINT:P
RINT:PRINT:FOR I=1 TO 31:PRINT CHR$(1);
CHR$(65+V);:V=V+1:NEXTI
230 FOR I=32 TO 255:PRINTCHR$(I);:NEXTI
240 DEF FN A(X)=ASC(CHR$(X))*8
250 X=1:FOR J=1 TO 255
260 FOR I=0 TO 7
270 VPOKE FNA(X)+I,PEEK(&HC800+I+J*8)
280 NEXT I : X=X+1
290 NEXT J
300 FOR I=32 TO 255:PRINT CHR$(I);:NEXT
 I
```

Figura 6

```
C000  11F897   LD    DE,97F8
C003  0608     LD    B,08
C005  C5       PUSH  BC
C006  21F8C9   LD    HL,C9F8
C009  0608     LD    B,08
C00B  1A       LD    A,(DE)
C00C  17       RLA
C00D  CB16     RL    (HL)
C00F  23       INC   HL
C010  10FA     DJNZ  C00C
C012  C1       POP   BC
C013  13       INC   DE
C014  10EF     DJNZ  C005
C016  C9       RET
```

Figura 7

```
10 REM IMPRESSOR DE
20 REM ALFABETOS
30 REM
40 REM PARA MSX
50 REM
60 REM GUILHERME A.L. DA SILVA
70 REM GUARARAPES-SP 09/09/88
80 REM
90 IF PEEK (&HE000)=1 THEN RESTORE 510:
GOTO 250
100 DATA 3E,00,32,F0,C4,FE,13,C2
110 DATA 0D,C0,C3,22,C0,FE,1B,20
120 DATA 06,3E,01,32,FF,C4,C9,3A
130 DATA FF,C4,FE,01,CA,2B,C0,3A
140 DATA F0,C4,CD,A5,00,3E,00,32
150 DATA FF,C4,C9,CD,A8,00,28,FB
160 DATA CD,B7,00,D8,21,00,C8,3A
170 DATA F0,C4,47,16,00,1E,08,19
180 DATA 10,FD,22,50,C1,E5,3E,1B
190 DATA CD,A5,00,3E,4B,CD,A5,00
200 DATA 3E,08,CD,A5,00,3E,00,CD
210 DATA A5,00,E1,06,FF,7E,22,52
220 DATA C1,CD,A5,00,D8,23,10,F5
230 DATA 21,00,C8,C9
240 FOR I=&HC000 TO &HC06B:READ V$:POKE
 I,VAL("&H"+V$):NEXT:POKE &HE000,1
250 REM DEFINE PARAMETROS
260 POKE &HC4FF,0:'DESLIGA GRAFICO
270 ES=1:'ESC
280 DEFUSR=&HC000:'DEFINE USR
290 POKE &HFBB0,255:'INTERRUPÇAO
300 POKE &HF417,1:'PADRAO ABNT
310 POKE &HC4F0,0:'CARACTERE INPUT
320 VB=&HC051:POKE VB,8:'PADRAO
330 VE=&HC05C:POKE VE,8:'DE PONTOS
340 REM PROGRAMA
350 READ C$: FOR I=1 TO LEN (C$):A$=MID
$(C$,I,1):IF A$="*" THEN POKE &HC001,27
:GOTO 400
360 IF A$="+" THEN POKE &HC001,19:GOTO
400
370 IF A$="%" THEN LPRINT CHR$(10):GOTO
 500
380 IF A$=CHR$(1) THEN I=I+1:H=ASC(MID$
(C$,I,1))-64:POKE&HC001,H:GOTO 400
390 POKE &HC001,ASC(A$)
400 A=USR(0):V=PEEK(&HC4F0):V2=PEEK(&HC
4FF)
410 IF INKEY$=CHR$(27) THEN ES=ES+1
420 IF ES/2<> INT(ES/2) THEN 500
430 REM REGISTRADORES
440 PRINT TAB(10"CARACTERE:";CHR$(V);"_
__Nº";V
450 PRINT TAB(10);STRING$(21,"-"):PRINT
 TAB(10)"GRAFICU";:IF PEEK(&HC4FF)<>1 T
HEN PRINT" DESLIGADO...";USING"^";V2:EL
SE PRINT" LIGADO ......."USING"^";V2
460 Z$=HEX$(PEEK(&HC151))+HEX$(PEEK(&HC
150))
470 PRINT TAB(10);"INIC HL ......";Z$
480 Z1$=HEX$(PEEK(&HC153))+HEX$(PEEK(&H
C152))
490 PRINT TAB(10);"INC HL  ......";Z1$
500 NEXT:LPRINT CHR$(13)
510 DATA"*AGUIA INFORMATICA+     AGUIA
INGORMATICA
```

Figura 8

```
C000  3E00      LD   A,00
C002  32F0C4    LD   (C4F0),A
C005  FE13      CP   13
C007  C20DC0    JP   NZ,C00D
C00A  C322C0    JP   C022
C00D  FE1B      CP   1B
C00F  2006      JR   NZ,C017
C011  3E01      LD   A,01
C013  32FFC4    LD   (C4FF),A
C016  C9        RET
C017  3AFFC4    LD   A,(C4FF)
C01A  FE01      CP   01
C01C  CA2BC0    JP   Z,C02B
C01F  3AF0C4    LD   A,(C4F0)
C022  CDA500    CALL 00A5
C025  3E00      LD   A,00
C027  32FFC4    LD   (C4FF),A
C02A  C9        RET
C02B  CDA800    CALL 00AB
C02E  28FB      JR   Z,C02B
C030  CDB700    CALL 00B7
C033  D8        RET  C
C034  2100C8    LD   HL,C800
C037  3AF0C4    LD   A,(C4F0)
C03A  47        LD   B,A
C03B  1600      LD   D,00
C03D  1E08      LD   E,08
C03F  19        ADD  HL,DE
C040  10FD      DJNZ C03F
C042  2250C1    LD   (C150),HL
C045  E5        PUSH HL
C046  3E1B      LD   A,1B
C048  CDA500    CALL 00A5
C04B  3E4B      LD   A,4B
C04D  CDA500    CALL 00A5
C050  3E08      LD   A,08
C052  CDA500    CALL 00A5
C055  3E00      LD   A,00
C05A  E1        POP  HL
C05B  0608      LD   B,08
C05D  7E        LD   A,(HL)
C05E  2252C1    LD   (C152),H
C061  CDA500    CALL 00A5
C064  D8        RET  C
C065  23        INC  HL
C066  10F5      DJNZ C05D
C068  2100C8    LD   HL,C800
C06B  C9        RET
```

2.1 UTILIZAÇÃO

2.1.1 Rotina I

Para funcionar, é necessário que se carregue um alfabeto do Graphos III. Se sua versão é a 1.1, mude na linha 130 o valor da variável H de 9200H para 9000H. Rode o programa. Automaticamente, este "tombará" os bytes e os deslocará para os endereços C800H à D000H. Na tela aparecerão os endereços que estão sendo lidos e os que estão sendo tombados. Após isso, redefinirá os caracteres para testar a operação. Grave com: BSAVE"INVTXT-1.ASM",&HC800,&HD000

2.1.2 Estrutura

LINHAS	FUNÇÃO
10-80	Linhas REM
81	Coloca a rotina Assembly na memória
90	CLS
100	Loop principal
110-120	Chama assembly
130	Define o endereço da tabela de bytes do alfabeto
140-150	Indicam à L.M. onde encontrar os bytes do alfabeto
160	Define o endereço onde os bytes "tombados" serão colocados
170-180	Indicam à L.M. onde colocar os bytes
190	Fecha o loop
191	Espera uma tecla
195	Redefine
210	SCREEN 1
220-230	Imprime set de caracteres MSX
250-290	Loop de redefinição
300	Imprime parcialmente o novo set

2.1.3 Variáveis

H	Início do alfabeto fonte
G	Representação hexadecimal de H
G$	Representação hexadecimal de G
FN a(x)	Endereço do caracter na tabela de padrões da VRAM

Carregue os dados invertidos na memória com BLOAD"INVTXT-1.asm". Digite o programa impressor e rode. Enquanto estiver imprimindo, aperte ESC e verá, num passe de mágica, os misteriosos conteúdos dos registradores do Z80, HL (endereço dos bytes) e A (código ASCII do caracter), além de saber se o modo gráfico está ligado ou não.

Na linha 510 existe um comando DATA para colocar mensagens.

Existem 3 caracteres de controle. São eles:

- Caracter * (asterisco) - liga o modo gráfico
- Caracter + (sinal mais) - desliga o modo gráfico
- Caracter % (porcentagem) - dá um line feed na impressora

Nas linhas 320 e 330 é definido o comprimento do caracter (pontos na horizontal), valores de 1 a 8.

Para interromper, aperte:

CONTROL + SHIFT + LGRA + RGRA.

Evite o CONTROL + STOP.

2.2.2 Estrutura

LINHAS	FUNÇÃO
10-80	Linhas REM
90	Verifica se precisa colocar na memória a rotina em L.M.
10-240	Rotina em L.M.
250-330	Inicializa
350-370	Lê mensagem e define caracteres de controle
380	Converte caracteres gráficos e indica a L.M.
390	Indica `a L.M. caracter comum
400	Chama L.M. e lê registradores do Z80
410-420	Espera ESC
440-490	Mostra conteúdo dos registradores Z80
500	Fim
510	DATA da mensagem

2..3 Variáveis

C$	String da mensagem
A$	String da letra a imprimir
V	Código ASCII do caracter
V"	Indicador do modo gráfico
ES	Indicador da tecla ESC
Z$	HL antes da impressão
Z1$	HL depois da impressão

3. CONVERSOR

O programa tem como objetivo converter alfabetos do Graphos (abreviatura Gra.) para o Print-X-Press (abreviatura P-X-P), que é uma Page Maker, ou vice-versa. As vantagens são claras: o Graphos tem um editor potente, fácil e com uma gama de 28 alfabetos disponíveis no mercado; o P-X-P tem um editor super potente (tem recursos de animação por "filmation", imprime os caracteres em um quadriculado na impressora com os bytes em hexadecimal, decimal e binário; fazendo, também, caracteres coloridos), mas não é de fácil utilização, além de ter um compilador que é uma "mão na roda".

Então, esse conversor é útil para os felizardos que possuem os dois programas.

3.1 Funcionamento

Opção 1	Permite ler um alfabeto do P-X-P, com a extensão .MVF (P-X-P)
Opção 2	Lê arquivos com deslocamento 1 (P-X-P/GRA) e 2 (GRA/P-X-P)
Opção 3	Grava com parâmetros 1 (GRA/P-X-P) e 2 (P-X- P)/GRA).
Opção 4	Cria partícula .DAT, usado só para GRA/P-X-P.
Opção 5	Mostra tabela de parâmetros
Opção 6	Files
Opção 7	Fim do programa

Obs: Quando se quer utilizar o conversor do GRA para o P-X-P, deve-se adotar 1 como deslocamento e parâmetro e, quando se quer o contrário, adota-se 2.

3.2 Como utilizar

A conversão do GRA/P-X-P é feita do seguinte modo:

- pressione 1	carregue um arquivo .MVF;
- pressione 2	carregue um alfabeto. ALF com deslocamento 1;
- pressione 3	grave um arquivo com extensão .MVF com parâmetro 1;
- pressione 4	dê o nome do arquivo acima gravado - agora sem extensão.

Do P-X-P/GRA é a seguinte:

- pressione 2	carregue um arquivo.MVF e com deslocamento 2
- pressione 3	grave um arquivo .ALF com parâmetro 2

Obs: MVF é extensão do P-X-P; ALF é extensão do Graphos.

3.3 Estrutura

LINHAS	FUNÇÃO
10-19	Linhas REM
20-23	Inicializa variáveis
30-100	Menu
110-120	Seleciona escolha
130	Volta ao menu
140	Fim
150	Opção 1
154	Recolhe dados
155	Lê arquivo .MVF (P-X-P)
158-199	Retorna
200	Opção 2
210-230	Recolhe dados
240	Lê arquivo D$ com o deslocamento D%(DE%)
245-299	Retorna
250	Opção 3
252-258	Recolhe dados
260	Grava arquivo G$ iniciando em EI% (P%) e finalizando EF%(P%)
270-299	Retorna
300	Opção 4
310-320	Recolhe dados
330-345	Grava arquivo .DAT
346-349	Retorna
350	Opção 5
352-358	Imprime tabela
398-399	Retorna
400	Opção 6
410-430	Diretório
500-550	Rotina de tratamento de erros

3.4 Variáveis

ES()	Matriz dos parâmetros
EI%()	Matriz do endereço inicial
EF%()	Matriz do endereço final
D%()	Matriz do deslocamento
P%	Parâmetro escolhido
DE%	Deslocamento escolhido
S$	Escolha
PX$	Nome do arquivo na opção 1
D$	Nome do arquivo na opção 2
G$	Nome do arquivo na opção 3
PT$	Nome do arquivo na opção 4

```
10 REM
11 REM CONVERSOR
12 REM DE ALFABETOS
13 REM
14 REM PARA MSX 1.1
15 REM
16 REM GUILHERME A. L. SILVA
17 REM 02-09-88
18 REM GUARARAPES
19 REM
20 KEYOFF:CLEAR 200,&H9200:ON ERROR GOT
0 500
21 DATA D0FC,9200,DA10,99FF,3EFC,C104
22 FOR I%=1 TO 6: READ E$(I%):NEXT
23 FOR I%=1TO2: EI%(I%)=VAL("&H"+E$(I%)
) : EF%(I%)=VAL("&H"+E$(I%+2)):D%(I%)=V
AL("&H"+E$(I%+4)):NEXT
30 CLS:PRINT TAB(5);"# TRANSFORMADOR DE
 ALFABETOS #"
40 LOCATE 15,3:PRINT"MENU"
50 LOCATE 10,6:PRINT"[1] LER P-X-P"
60 LOCATE 10,8:PRINT"[2] LER ALFABETO"
70 LOCATE 10,10:PRINT"[3] GRAVAR ALFABE
TO"
80 LOCATE 10,12:PRINT"[4] CRIAR PARTICU
LA"
90 LOCATE 10,14:PRINT"[5] TABELA"
95 LOCATE 10,16:PRINT"[6] FILES"
96 LOCATE 10,18:PRINT"[7] BASIC"
100 LOCATE 15,21:PRINT"OPCAO: ";:S$=INP
UT$(1)
110 IF S$<"1" OR S$>"7" THEN 100
120 ON VAL(S$) GOSUB 150,200,250,300,35
0,400,140
130 GOTO 30
140 CLS:KEYON:END
150 CLS:PRINTSPC(14);"OPCAO 1:":PRINT:P
RINT SPC(6);"LER ARQUIVO PRINT-X-PRESS"
152 LOCATE 11,9:PRINT"NOME DO ARQUIVO:
":LOCATE 11,10:INPUT PX$
154 LOCATE 6,13:PRINT"COLOQUE O DISCO N
O DRIVE A":A$=INPUT$(1)
156 BLOAD PX$
158 LOCATE 18,21:PRINT"OK":A$=INPUT$(1)
199 RETURN
200 CLS:PRINT SPC(14);"OPCAO 2":PRINT:P
RINT SPC(4);"LER ARQUIVO COM DESLOCAMEN
TO"
210 LOCATE 9,6:PRINT"NOME DO ARQUIVO: "
:LOCATE 9,7:INPUT D$
220 LOCATE 9,11:PRINT"DESLOCAMENTO (1-2
)":LOCATE 9,12:INPUT DE%:IF DE% <> 1 AN
D DE% <> 2 THEN 220
230 LOCATE 6,16:PRINT"COLOQUE O DISCO N
O DRIVE A":A$=INPUT$(1)
240 BLOAD D$,D%(DE%)
245 LOCATE 18,21:PRINT "OK":A$=INPUT$(1
)
249 RETURN
250 CLS:PRINT SPC(15);"OPCAO 3: ":PRINT
:PRINT SPC(11);"GRAVAR ARQUIVO"
252 LOCATE 11,6:PRINT"NOME DO ARQUIVO:
":LOCATE 11,7:INPUT G$
254 LOCATE 11,11:PRINT"PARAMETROS (1-2)
: ":LOCATE 11,12:INPUT P%:IF P%<1 OR P%
>2 THEN 254
258 LOCATE 2,15:PRINT"COLOQUE O DISCO D
ESTINO NO DRIVE A":A$=INPUT$(1)
260 BSAVE G$,EI%(P%),EF%(P%)
270 LOCATE 18,21:PRINT"OK":A$=INPUT$(1)
299 RETURN
300 CLS:PRINT SPC(15);"OPCAO 4:":PRINT:
PRINT SPC(9);"GRAVAR PARTICULA.DAT"
310 LOCATE 11,8:PRINT"NOME DO ARQUIVO:
":LOCATE 11,9:INPUT PT$
320 LOCATE 6,14:PRINT"COLOQUE O DISCO N
O DRIVE A":A$=INPUT$(1)
330 OPEN PT$+".DTA" FOR OUTPUT AS #1
340 PRINT #1,PT$+".MVF"
345 CLOSE #1
346 LOCATE 18,21:PRINT"OK":A$=INPUT$(1)
349 RETURN
350 COLOR 3:CLS:PRINT SPC(15):"OPCAO 5:
":PRINT:PRINT SPC(9):"TABELA DE PARAMET
ROS"
352 LOCATE 0.5:PRINT STRING$(39,"-"):LO
CATE 0.6:PRINT SPC(4)"PARAMETRO 1";SPC(
9);"PARAMETRO 2":LOCATE 0.7:PRINT STRIN
G$(39,"-")
354 FOR I%=0 TO 10:LOCATE 18,8+I%:PRINT
" ":NEXT:LOCATE 0,18:PRINT STRING$(39,"
-"):LOCATE 0,20:PRINT STRING$ (39,"-")
356 LOCATE 3,9:PRINT"INICIO: D0FC":LOCA
TE 3,12:PRINT"FINAL: DA10":LOCATE 3,15:
PRINT"DESLOC: 3EFC"
358 LOCATE 23,9:PRINT"INICIO: 9200": LO
CATE 23,12:PRINT"FINAL: 99FF":LOCATE 23
,15:PRINT"DESLOC: C104"
360 LOCATE 17,19:PRINT"OK":A$=INPUT$(1)
398 COLOR 15,1,1
399 RETURN
400 CLS:PRINTSPC(15);"OPCAO 6: ":PRINT:
PRINTSPC(10)"ARQUIVOS DO DISCO"
410 PRINT:FILES
420 PRINT:PRINT:PRINTSPC(17)"OK":A$=INP
UT$(1)
430 RETURN
500 IF ERR=53 THEN LOCATE 10,19:PRINT"A
RQUIVO INEXISTENTE":PLAY"V1507C4":RESUM
E NEXT
510 IF ERR=56 THEN LOCATE 13,19:PRINT"N
OME ILEGAL":PLAY "V1507C4":RESUME NEXT
520 IF ERR=69 THEN LOCATE 13,19:PRINT"e
rro de e/s":PLAY"v1507c4":RESUME NEXT
530 IF ERR=68 THEN LOCATE 11,19:PRINT"D
ISCO PROTEGIDO":PLAY"V1507C4":RESUME NE
XT
540 IF ERR=66 THEN LOCATE 13,19:PRINT"D
ISCO CHEIO":PLAY"V1507C4":RESUME NEXT
550 LOCATE 15,19:PRINT"ERRO ";ERR:PLAY"
V1507C4":RESUME:NEXT
```


* DRIVE 5.1/4 SLIN COMPLETO
* PLACA 80 COLUNAS
* MODEN DE COMUNICAÇÃO
* EXPANSOR DE SLOT (C/4 SLOTS)
* GABINETE P/DRIVE COM FONTE FRIA
* INTERFACE DUPLA P/DRIVE

* PACOTÃO EM DISCO: 100 JOGOS (ESCOLHER) + 5 APLICATIVOS + 10 DISCOS = 60.00
* PACOTÃO EM FITA: 100 JOGOS (ESCOLHER) + 5 APLICATIVOS + 7 FITAS = 60.00

SOLICITE NOSSO CATÁLOGO DE PROGRAMAS PARA FAZER A SUA ESCOLHA. ATENDEMOS TODOS OS ESTADOS EM 24 HORAS VIA SEDEX
PARA FAZER SEU PEDIDO ENVIE CHEQUE NOMINAL COM CARTA DETALHADA PARA **A. NASSER.**
RIO DE JANEIRO: RUA GONZAGA BASTOS 411/203 — VILA ISABEL — RJ CEP 20541 — TEL (021) 234 0775
FILIAL CURITIBA; AV 7 DE SETEMBRO. 3146 LOJA 20 — SHOPPING SETE — CURITIBA — PR — CEP 80010 — TEL 232-0399

ABASTEÇA O SEU MSX NA ECTRON.

EQUIPAMENTOS PARA MSX

Driver MSX 5 1/4
Driver MSX 3 1/2
Video Station
Interface p/ Drive
Cartão 80 Colunas
Modem
Monitores de Vídeo
Gabinete e Fonte p/ Driver
Ponta Discos "Acrílico" 100 Discos
Mesa para Computador
Mesa para Impressora

— Fitas de Vídeo

• Na Ecatron você encontra o último lançamento "MPO" em videocassete "Curso de Basic MSX" acompanha livro.
• "Dominando e MSX"

SOFTWARE

• D.Base Ferramenta Profissional p/ manipulação de banco de dados.
• Super Calc: A mais famosa Planilha de cálculos.
(Ambos com suporte técnico e reposição de versão)

SUPRIMENTOS

Disquetes
Fitas p/ Impressora
Formulários Contínuos
Capas Protetoras p/ Equipamentos

APLICATIVOS

Os mais potentes do mercado

LITERATURAS

Livros 100 Dicas
Livros 50 Dicas (EM LANÇAMENTO)
Livros Programação Avançada
Livros Astrologia
Livros Curso de Música
Livros Curso de Basic

JOGOS

Temos a coleção completa inclusive os últimos lançamentos.

Solicite nosso catálogo inteiramente GRÁTIS.

* Nossa caixa Postal 12005 - Cep. 02098/ São Paulo/

ECTRON ELETRÔNICA LTDA.

Rua Dr. Cesar, 131 - Metrô Santana - S.Paulo/SP

TEL.: (011) 290-7266

GTM&P

BUTTY SPRITE

ALEXANDRE C. DULLIUS

O programa que apresento é um editor de sprites com vários comandos que irão auxiliá-lo na confecção de sprites de 16x16.

Este programa chama-se Butty Sprite e a sua versão é a 1.0. Sua operação é simples e a sua digitação um pouco cansativa.

Depois de digitado e gravado em disquete ou fita cassete, rode o programa com o comando RUN.

Para mover o sprite dentro do quadro de desenho, use as setas que movimentam o cursor e, para marcar ou apagar um ponto do sprite, use a barra de espaços.

Para ter acesso aos comandos, deve-se teclar, primeiramente, a tecla F1.

Os comandos disponíveis no Butty Sprite são os seguintes:

HOME/CLS - Limpa o sprite, apagando-o da memória e da tela.

INSERT - Pinta toda uma linha.

DELETE - Apaga toda uma linha.

ESC - Sai do programa. Nesta opção, o programa irá solicitar confirmação e, em caso positivo, o programa acessa o Basic.

CONTROL + A - Inverte todo o sprite.

CONTROL + B - Faz com que todo o sprite seja deslocado uma linha para cima.

CONTROL + C - Faz com que todo o sprite seja deslocado uma linha para baixo.

CONTROL + D - Faz com que todo o sprite seja deslocado uma coluna para a direita.

CONTROL + E - Faz com que todo o sprite seja deslocado uma coluna para a esquerda.

CONTROL + F - Faz uma cópia do bloco 1 sobre os blocos 2, 3 e 4. O conteúdo dos blocos 2, 3 e 4 é perdido.

CONTROL + G - Faz uma cópia do bloco 1 sobre o 3 e do bloco 2 sobre o 4 . O conteúdo dos blocos 3 e 4 é perdido.

CONTROL + H - Lista estes comandos no vídeo.

CONTROL + I - Lista estes comandos na impressora.

CONTROL + J - Lista os dados do sprite em hexadecimal na tela.

CONTROL + M - Lista os dados do sprite em hexadecimal na impressora.

CONTROL + N - Inverte coluna.

CONTROL + O - Inverte linha.

Para ler o sprite, digite este pequeno programa:

```
10 SCREEN 2,2 : FOR A=1 TO 16 : READ A$,B$20
C$=C$+CHR$(VAL("&H"+A$))
30D$=D$+CHR$(VAL("&H"+B$))40 NEXT A :
SPRITE$(1)=C$+D$50 PUTSPRITE0,(127,97),15,1
: GOTO 50
60DATAFF,FF,80,1,80,1,80,1,80,1,80,1,80,1,80,
1,80,1,80,1,80,1,80,1,80,1,80,1,80,1,FF,FF
```

```
1 ' *********************************
2 ' *     BUTTY-SPRITE's 16x16      *
3 ' * By:Alexandre Cardoso Dullius  *
4 ' *    Rua Lindolfo Collor 111    *
5 ' *    Sapiranga  RS CEP:93800    *
6 ' * Revista CPU  "guia Inf. LTDA  *
7 ' *********************************
8 FORA=1TO10:KEYA,"":NEXTA:GOTO65
9 ' CTRL+A
10 GOSUB95:A$="CTRL+A":GOSUB62:GOSUB122
:L=0:FORC=41TO162STEP8:K=0:L=L+1:FORD=1
6TO138STEP8:K=K+1
11 IFPOINT(D+5,C+5)=14THENLINE(D+1,C+1)
-(D+8,C+8),1,BF:PRESET(K+17,L+17),1:P(K
,L)=0:GOTO13
12 LINE(D+1,C+1)-(D+8,C+8),14,BF:PRESET
(K+17,L+17),15:P(K,L)=1
13 NEXTD,C
14 GOTO88
15 ' ESC
16 GOSUB95:A$="ESC":GOSUB62:E=155:F=106
:G=15:A$="CONFIRMA(S/N)":GOSUB62
17 B$=INPUT$(1):IFB$="S"THENEND
18 IFB$="N"THEN88
19 GOTO17
20 ' DELETE
21 GOSUB95:A$="DELETE":GOSUB62:FORD=16T
O138STEP8:LINE(D+1,Y+1)-(D+8,Y+8),1,BF:
NEXTD
22 C=INT((Y-40)/8)+1:FORZ=1TO16:P(Z,C)=
0:PRESET(Z+17,C+17),1:NEXTZ
23 GOTO88
24 ' INSERT
25 GOSUB95:A$="INSERT":GOSUB62:FORD=16T
O138STEP8:LINE(D+1,Y+1)-(D+8,Y+8),14,BF
:NEXTD
26 C=INT((Y-40)/8)+1:FORZ=1TO16:P(Z,C)=
1:PRESET(Z+17,C+17),15:NEXTZ
27 GOTO88
28 ' HOME OU CLS
29 GOSUB95:A$="HOME/CLS":GOSUB62:GOSUB1
22
30 LINE(14,14)-(40,38),1,BF:LINE(14,14)
-(40,38),2,B:LINE(15,40)-(145,170),1,BF
:LINE(15,40)-(145,170),4,B:FORD=1TO16:F
ORC=1TO16:P(D,C)=0:NEXTC,D:GOTO88
31 ' F1
32 SOUND8,8:FORA=255TO1STEP-15:SOUND0,A
:NEXTA:SOUND0,0:LINE(90,180)-(150,190),
1,BF:LINE(152,100)-(240,188),1,BF:LINE(
152,100)-(240,188),6,B
33 A$=INPUT$(1):IFASC(A$)=32THEN33
34 IFASC(A$)=11ORASC(A$)=12THEN29
35 IFASC(A$)=18THEN25
```

```
36 IFASC(A$)=127THEN21
37 IFASC(A$)=27THEN16
38 D=ASC(A$):ONDGOTO10,97,90,111,107,11
4,118,124,147,151,,,161,166,171
39 GOTO33
40   ACENDE OU APAGA
41 IFSTRIG(0)<>0ANDPOINT(X+5,Y+5)=14THE
NLINE(X+1,Y+1)-(X+8,Y+8),1,BF:D=INT((X-
15)/8)+1:C=INT((Y-40)/8)+1:P(D,C)=0:PRE
SET(D+17,C+17),1:GOTO43
42 IFSTRIG(0)<>0THENLINE(X+1,Y+1)-(X+8,
Y+8),14,BF:D=INT((X-15)/8)+1:C=INT((Y-4
0)/8)+1:P(D,C)=1:PRESET(D+17,C+17),15
43 RETURN
44 PUTSPRITE0,(X,Y),15,1:GOSUB41
45 GOTO65
46 ' MOVIMENTO
47 K=STICK(0)+STICK(1)+STICK(2)
48 IFK=1THENY=Y-8
49 IFK=2THENX=X+8:Y=Y-8
50 IFK=3THENX=X+8
51 IFK=4THENX=X+8:Y=Y+8
52 IFK=5THENY=Y+8
53 IFK=6THENX=X-8:Y=Y+8
54 IFK=7THENX=X-8
55 IFK=8THENX=X-8:Y=Y-8
56 IFX<16THENX=16
57 IFX>136THENX=136
58 IFY<41THENY=41
59 IFY>161THENY=161
60 RETURN
61 ' IMPRESSAO
62 FORD=1TOLEN(A$):SOUND0,E:DRAW"BM"+ST
R$(E)+","+STR$(F):COLORG:PRINT#1,MID$(A
$,D,1)
63 E=E+H:NEXTD:SOUND0,0:RETURN
64 ' Rotina Defaut
65 COLOR15,1,1:CLEAR:DEFINTA-Z:MAXFILES
=3:OPEN"GRP:"FOROUTPUTAS#1:SCREEN2,2,0:
BEEP
66 ' Limpa matriz,define variaveis
67 DIMP(16,16),J(16,16):FORC=1TO16:FORD
=1TO16:P(C,D)=0:NEXTD,C:X=16:Y=41:S=1:B
EEP
68 ' Define Sprite
69 FORD=1TO8:READA$:B$=B$+CHR$(VAL("&H"
+A$)):NEXTD:BEEP:SPRITE$(1)=B$:DATA81,6
6,66,0,0,66,66,81
70 POKE&HFCAB,1
71 ' DESENHA TELA
72 LINE(14,14)-(40,38),2,B
73 LINE(15,40)-(145,170),4,B
74 E=70:F=3:G=10:H=7:A$="BUTTY-SPRITE":
SOUND8,8
75 GOSUB62:E=71:GOSUB62:E=44:F=13:G=3:H
=7:A$="By->Alexandre Cardoso Dullius":G
OSUB62
76 E=43:GOSUB62:E=55:F=23:G=7:H=16:A$="
REVISTA CPU":GOSUB62:E=56:GOSUB62
77 E=63:F=32:G=10:H=7:A$="Aguia Informa
tica LTDA":GOSUB62:E=62:GOSUB62
78 E=152:F=50:G=15:H=6:A$="PARA DIGITAR
 UM":GOSUB62:E=152:F=58:A$="COMANDO , T
ECLE":GOSUB62
79 E=152:F=66:A$="  F1 ANTES !!!":GOSUB
62
80 E=28:F=181:G=11:H=7:A$="COMANDO=>":G
OSUB62
81 LINE(152,100)-(240,188),6,B:BEEP
82 ' LOOP PRINCIPAL
83 PUTSPRITE0,(X,Y),15,1:GOSUB47
84 PUTSPRITE0,(X,Y),15,1:GOSUB41
85 KEY(1)ON:ONKEYGOSUB32
86 GOTO83
87 ' LIMPA QUADRO
88 LINE(152,100)-(240,188),1,BF:LINE(15
2,100)-(240,188),6,B:GOTO83
89 ' CTRL+C
90 GOSUB95:A$="CTRL+C":GOSUB62:GOSUB122
91 FORB=1TO16:J(B,1)=P(B,16):NEXTB
92 FORA=1TO15:FORB=1TO16:J(B,A+1)=P(B,A
):NEXTB,A
93 GOTO102
94 ' DDD
95 E=90:F=181:G=10:H=6:RETURN
96 ' CTRL+B
97 GOSUB95:A$="CTRL+B":GOSUB62:GOSUB122
98 FORB=1TO16:J(B,16)=P(B,1):NEXTB
99 FORA=16TO2STEP-1:FORB=1TO16:J(B,A-1)
=P(B,A):NEXTB,A
100 GOTO102
101   AJUDA
102 FORA=1TO16:FORB=1TO16:P(B,A)=J(B,A)
:NEXTB,A:K=0:FORB=41TO163STEP8:K=K+1:L=
0
103 FORA=16TO137STEP8:L=L+1:IFP(L,K)=1T
HENLINE(A+1,B+1)-(A+8,B+8),14,BF:PRESET
(L+17,K+17),15:GOTO105
104 LINE(A+1,B+1)-(A+8,B+8),1,BF:PRESET
(L+17,K+17),1
105 NEXTA,B:GOTO88
106 ' CTRL+E
107 GOSUB95:A$="CTRL+E":GOSUB62:GOSUB12
2
108 FORB=1TO16:J(16,B)=P(1,B):NEXTB
109 FORA=16TO2STEP-1:FORB=1TO16:J(A-1,B
)=P(A,B):NEXTB,A:GOTO102
110 ' CTRL+D
111 GOSUB95:A$="CTRL+D":GOSUB62:GOSUB12
2
112 FORB=1TO16:J(1,B)=P(16,B):NEXTB:FOR
A=1TO15:FORB=1TO16:J(A+1,B)=P(A,B):NEXT
B,A:GOTO102
113 ' CTRL+F
114 GOSUB95:A$="CTRL+F":GOSUB62:GOSUB12
2
115 FORA=1TO16:FORB=1TO16:J(B,A)=P(B,A)
:NEXTB,A
116 FORA=2TO4:FORB=1TO16:J(A,B)=P(1,B):
NEXTB,A:GOTO102
117 ' CTRL+G
118 GOSUB95:A$="CTRL+G":GOSUB62:GOSUB12
2
119 FORA=1TO16:FORB=1TO16:J(B,A)=P(B,A)
```

```
:NEXTB,A
120 FORB=1TO16:J(3,B)=P(1,B):J(4,B)=P(2
,B):NEXTB:GOTO102
121 ' PROCESSANDO
122 E=155:F=123:G=15:H=6:A$="AGUARDE!":
GOSUB62:E=160:F=148:G=14:H=6:A$="PROCES
SANDO...":GOSUB62:RETURN
123 ' CTRL+H
124 RESTORE126:SCREEN2:G=11:F=5:H=6:FOR
B=1TO17:E=9:READA$:GOSUB62:LINE(6,F-2)-
((LEN(A$)*6)+13,F+8),13,B:F=F+11:NEXTB
125 A$=INPUT$(1):CLS:K=0:FORB=41TO162ST
EP8:K=K+1:L=0:FORA=16TO137STEP8:L=L+1:G
OTO143
126 DATA HOME/CLS - LIMPA A SPRITE
127 DATA INSERT - PINTA TODA LINHA
128 DATA DELETE - AFAGA TODA LINHA
129 DATA ESC - SAI DO PROGRAMA
130 DATA CTRL+A - INVERTE OS QUADROS
131 DATA CTRL+B - SCROLL P/ CIMA
132 DATA CTRL+C - SCROLL P/ BAIXO
133 DATA CTRL+D - SCROLL P/ DIREITA
134 DATA CTRL+E - SCROLL P/ ESQUERDA
135 DATA CTRL+F - ESPELHO 1/4
136 DATA CTRL+G - ESPELHO 1/2
137 DATA CTRL+H - LISTA COMANDOS VIDEO
138 DATA CTRL+I - LISTA COMANDOS IMPRES
SORA
139 DATA CTRL+J - LISTA DADOS HEX. VIDE
O
140 DATA CTRL+M - LISTA DADOS HEX. IMPR
ESSORA
141 DATA CTRL+N - INVERTE SO COLUNA
142 DATA CTRL+O - INVERTE SO LINHA
143 IFP(L,K)=1THENLINE(A+1,B+1)-(A+8,B+
8),14,BF:PRESET(L+17,K+17),15:GOTO145
144 LINE(A+1,B+1)-(A+8,B+8),1,BF:PRESET
(L+17,K+17),1
145 NEXTA,B:RETURN72
146 ' CTRL+I
147 GOSUB95:A$="CTRL+I":GOSUB62:E=153:F
=105:G=15:H=6:A$="TECLE ALGO":GOSUB62:E
=158:F=113:A$="PARA IMPRIMIR":GOSUB62
148 A$=INPUT$(1):E=160:F=132:G=14:A$="I
MPRIMINDO...":GOSUB62:LPRINT"   COMANDO
S DO BUTTY SPRITE 1.0"
149 RESTORE126:BEEP:FORB=1TO17:READA$:L
PRINTA$:LPRINT:NEXTB:GOTO88
150 ' CTRL+J
151 GOSUB95:A$="CTRL+J":GOSUB62:GOSUB12
2:SCREEN2:COLOR15:DRAW"BM9,5":FORA=1TO1
6:FORB=1TO16
152 P$=P$+RIGHT$(STR$(P(B,A)),1):NEXTB
153 T$=LEFT$(P$,8):K$=RIGHT$(P$,8)
154 U$=HEX$(VAL("&B"+T$)):L$=HEX$(VAL("
&B"+K$))
155 PRINT#1,U$;",";L$;",";:P$="":T$=P$:
K$=T$:L$=K$
156 IFA=5THENDRAW"BM9,15"
157 IFA=10THENDRAW"BM9,25"
158 IFA=15THENDRAW"BM9,35"
159 NEXTA:GOTO125
160 ' CTRL+M
161 GOSUB95:A$="CTRL+M":GOSUB62:E=155:F
=123:G=14:H=6:A$="TECLE ALGO":GOSUB62:E
=160:F=133:G=15:A$="PARA IMPRIMIR":GOSU
B62
162 A$=INPUT$(1):E=155:F=145:A$="IMPRIM
INDO...":GOSUB62:LPRINT"  DADOS DO SPRI
TE...CRIADO NO BUTTY"
163 FORA=1TO16:FORB=1TO16:P$=P$+RIGHT$(
STR$(P(B,A)),1):NEXTB:T$=LEFT$(P$,8):K$
=RIGHT$(P$,8):U$=HEX$(VAL("&B"+T$))
164 L$=HEX$(VAL("&B"+K$)):LPRINTU$;",";
P$;",";:P$="":T$=P$:K$=T$:L$=K$:NEXTA:C
LS:GOTO125
165 ' CTRL+N
166 GOSUB95:A$="CTRL+N":GOSUB62:GOSUB12
2:FORB=41TO163STEP8:A=INT((X-15)/8)+1:C
=INT((B-40)/8)+1
167 IFP(A,C)=1THENP(A,C)=0:LINE(X+1,B+1
)-(X+8,B+8),1,BF:PRESET(A+17,C+17),1:GO
TO169
168 P(A,C)=1:LINE(X+1,B+1)-(X+8,B+8),14
,BF:PRESET(A+17,C+17),15
169 NEXTB:GOTO88
170 ' CTRL+O
171 GOSUB95:A$="CTRL+O":GOSUB62:GOSUB12
2:FORB=16TO138STEP8:A=INT((Y-40)/8)+1:C
=INT((B-15)/8)+1
172 IFP(C,A)=1THENP(C,A)=0:LINE(B+1,Y+1
)-(B+8,Y+8),1,BF:PRESET(C+17,A+17),1:GO
TO174
173 P(C,A)=1:LINE(B+1,Y+1)-(B+8,Y+8),14
,BF:PRESET(C+17,A+17),15
174 NEXTB:GOTO88
175 '______________________________
176 ' ESTE PROGRAMA FOI DESENVOLVIDO
177 ' POR ALEXANDRE CARDOSO DULLIUS,
178 ' EM DEZEMBRO DE 1988 , NUM MICRO
179 ' EXPERT,ESPECIALMENTE PARA A
180 ' REVISTA CPU.................
181 '______________________________
```

JANELAS EM SCREEN 0

RICARDO P. RYMSZA

Já ficou provado que a utilização de janelas, além de dar melhor apresentação aos programas ainda os torna de mais fácil utilazação. Porém, a programação de tais janelas era feita através do comando LINE, o que exigia que o programa trabalhasse em SCREEN 2, o que trazia um sério problema, que é a resolução de caracteres na tela.

O programa que apresento a seguir gera janelas de tamanhos diversos e em SCREEN 0, aproveitando a resolução de 40x24 caracteres, tendo possui duas variações de janelas: uma em que a janela se sobrepõe a outra anterior e outra em que a janela aparece e desaparece, mantendo o texto anterior intacto.

As linhas 500 a 580 salvam, a partir do endereço &HA000, a parte da tela que é apagada pela nova janela. Os caracteres são lidos diretamente da VRAM e escritos na RAM.

As linhas 700 a 780 recolocam na VRAM a parte da tela que foi apagada pela janela.

As linhas 1000 a 2050 geram as janelas definidas através dos números contidos nas variáveis X, Y, L e A.

Encerro este artigo com esta breve explicação do programa, lembrando ao leitor que nenhum programa é absoluto e nenhum programador é perfeito. O leitor deve modificar estas rotinas conforme as suas necessidades, aumentando a sua velocidade e aperfeiçoando-o, pois este é o único e verdadeiro modo se aprender a lidar com qualquer linguagem, desde o Basic ao Assembler. Qualquer dúvida ou sugestão será bem recebida, bastando para tal entrar em contato com a revista CPU.

```
1 ' Janelas em screen 0
2 ' por Ricardo P. Rymsza
3 ' Revista CPU
4 '
10 CLS
20 WIDTH40
29 KEYOFF
30 '
31 ' X e Y sao as coordenadas da janela
32 ' L e a largura da janela
33 ' A e a altura da janela
34 '
40 FORW=1TO880:PRINT"*";:NEXTW
60 X=2:Y=2:L=15:A=10:GOSUB1000
70 LOCATE3,3:PRINT"Esta rotina"
71 LOCATE3,4:PRINT"permite   a"
72 LOCATE3,5:PRINT"criacao  de"
73 LOCATE3,6:PRINT"Janelas  em"
74 LOCATE3,7:PRINT"Screen 0. "
75 LOCATE3,9:PRINT"Tecle espaco"
80 IFINKEY$=""THEN80
90 X=7:Y=5:L=15:A=10:GOSUB500:GOSUB1000
100 LOCATE8,6:PRINT"Cabe agora ao"
110 LOCATE8,7:PRINT"leitor, aper-"
120 LOCATE8,8:PRINT"feicoar  esta"
130 LOCATE8,9:PRINT"rotina ou "
140 LOCATE8,10:PRINT"criar novas."
150 LOCATE8,12:PRINT"Tecle espaco"
155 IFINKEY$=""THEN155
160 GOSUB700:X=2:Y=15:L=34:A=3:GOSUB100
0
170 LOCATE3,16:PRINT"Janelas v 1.2 por
Ricardo Rymsza"
190 GOTO 190
```

Ricardo P. Rymsza, 15, cursa a 1ª série do 2º grau do Colégio Santo Inácio, programando em micros da linha MSX há 3 anos, utilizando as linguagens Basic e Assembler.

MSX

"OS MAGOS DO MSX"

CONHEÇA NOSSA QUALIDADE IMPECÁVEL E A EXPERIÊNCIA DE QUEM LIDA COM MSX HA MAIS DE TRÊS ANOS. NOSSO ACERVO TEM CERCA DE 2100 TITULOS (MUITOS COM DOCUMENTAÇÃO), E CRESCE SEMANALMENTE COM GRANDES NOVIDADES E PROMOÇÕES...VAMOS! SEJA MAIS UM DOS MAIS DE MIL FELIZES CLIENTES DO MAGO DO LAZZAROSOFT.

JOGO	NCZ$1,00	APLICATIVO	NCZ$2,00
CP/M	NCZ$4,00	COPIADOR	NCZ$4,00
DISCO	NCZ$2,00	FITA	NCZ$2,00
DESPESAS	POSTAIS		NCZ$0,00

"MSXBOOK II (DICAS DE JOGOS)"
"EDTRONIC"

"FLINTSTONES•TRIPLE COMMANDO•SOL NEGRO 1 & 2 NAVY MOVIES•TERRA MEX•TUXY•TETRIS•WALL RUNNER•VECTOR MOON LANDING•ADDICTA BALL•NEO-Z•DANGER MOUSE• PERTER BEARDLEY'SFOOTBALL•QUEEN II•DUCKYS•FLICKY• SQUARE•RAMPART•CHUBBY CRISTLER•OPERATION WOLF• ASPAR•PITMAN•FINAL COUNTDOWN•RED OCTOBER ETC."

Seu pedido poderá ser pago com vale postal (AG. Centro) ou cheque nominal cruzado a Carlos H. B. Magalhães contendo um máximo de informações sobre seu equipamento e um telefone para eventual contato.

DESCONTÃO NO TOTAL: 10%
SOLICITE CATÁLOGO GRÁTIS
NOVIDADE SEM PACOTE É AQUI
ENTREGA EM 24 HORAS + CORREIO

CAIXA POSTAL 1955 RIO DE JANEIRO RJ CEP:20001
NOVO FONE: (021) 299-7080

```
500 '
501 ' Rotina que salva a parte da tela
502 ' que sera preenchida com a janela.
503 '
504 N=Y*40+X:O=&HA000
505 FORV=1TOA
510 R=N
520 FORF=1TOL
530 POKE(O),VPEEK(R)
540 O=O+1:R=R+1
550 NEXTF
560 N=N+40
570 NEXTV
580 RETURN
700 '
701 ' Rotina que retorna a parte da te-
702 ' la que foi preenchida c/ a janela
703 '
704 N=Y*40+X:O=&HA000
705 FORV=1TOA
710 R=N
720 FORF=1TOL
730 VPOKE(R),PEEK(O)
740 O=O+1:R=R+1
750 NEXTF
760 N=N+40
770 NEXTV
780 RETURN
1000 '
1001 ' Rotina que cria janelas em
1002 ' screen 0
1003 '
1004 IF(L+X)>39THENL=L-1:GOTO1004
1005 IF(A+Y)>23THENA=A-1:GOTO 1005
1006 W=Y:T=1:LOCATEX,Y
1010 PRINT"+";
1020 FORV=1TOL-2
1030 PRINT"-";:NEXTV
1040 PRINT"+"
1045 T=T+1
1050 IFT=ATHEN2000
1060 LOCATEX,W+1
1070 PRINT"!";
1080 FORR=X+1TO(X+(L-1))-1:LOCATER,W+1:
PRINT" ";:NEXTR
1100 LOCATER,W+1:PRINT"!"
1110 T=T+1:W=W+1:GOTO1050
2000 LOCATEX,Y+(A-1)
2010 PRINT"+";
2020 FORV=1TOL-2
2030 PRINT"-";:NEXTV
2040 PRINT"+"
2050 RETURN
```


ASSINE CPU POR 6 NÚMEROS

DESEJO EFETUAR A ASSINATURA DA REVISTA CPU POR 6 NÚMEROS

PARA TAL, ESTOU ENVIANDO CHEQUE NOMINAL A ÁGUIA INFORMÁTICA, OU VALE POSTAL (PAGÁVEL NA AGÊNCIA COPACABANA), NO VALOR DE NCz$ 7,00 (SETE CRUZADOS NOVOS).

FAÇA A SUA ASSINATURA E CONCORRA AO SORTEIO DE PROGRAMAS DA PAULI SOFT.
(EDTRONIC - SPRITE MAKER - MSX TURBO - FAST COPY)

NOME: ....

ENDEREÇO : ....

BAIRRO : .... CEP : ....

CIDADE : .... ESTADO : ....

DADOS DO EQUIPAMENTO : ....

....

AGUIA INFORMÁTICA

DESEJO EFETUAR A ASSINATURA DA REVISTA CPU POR 6 NÚMEROS

PARA TAL, ESTOU ENVIANDO CHEQUE NOMINAL A ÁGUIA INFORMÁTICA, OU VALE POSTAL (PAGÁVEL NA AGÊNCIA COPACABANA), NO VALOR DE NCz$ 7,00 (SETE CRUZADOS NOVOS).

FAÇA A SUA ASSINATURA E CONCORRA AO SORTEIO DE PROGRAMAS DA PAULI SOFT.
(EDTRONIC - SPRITE MAKER - MSX TURBO - FAST COPY)

NOME: ....

ENDEREÇO : ....

BAIRRO : .... CEP : ....

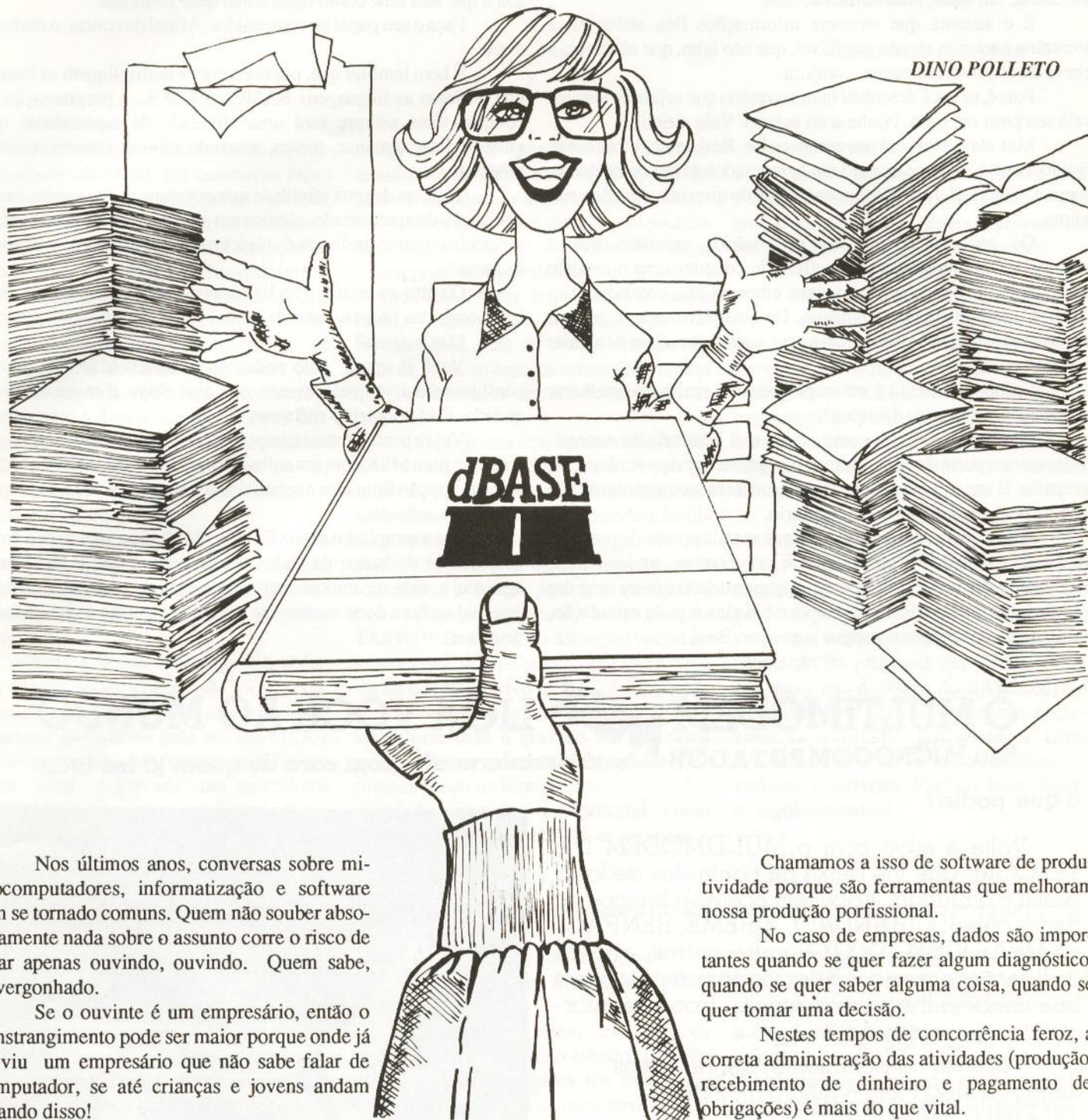
CIDADE : .... ESTADO : ....

DADOS DO EQUIPAMENTO : ....

....

AGUIA INFORMÁTICA

MSX E dBASE II, A CAMINHO DE SUA EMPRESA

DINO POLLETO

Nos últimos anos, conversas sobre microcomputadores, informatização e software têm se tornado comuns. Quem não souber absolutamente nada sobre o assunto corre o risco de ficar apenas ouvindo, ouvindo. Quem sabe, envergonhado.

Se o ouvinte é um empresário, então o constrangimento pode ser maior porque onde já se viu um empresário que não sabe falar de computador, se até crianças e jovens andam falando disso!

Mas, a verdade é que qualquer pessoa estará sendo injusta consigo mesma se considerar-se "ignorante" em computação. Afinal, quem fica envergonhado por não saber nada sobre geladeiras além, é claro, de abrir a porta e apanhar uma cerveja geladinha, ou por não saber nada sobre mecânica de automóveis?

Do mesmo modo, ninguém deveria se esconder cada vez que começasse a ouvir falar em teclados, micros, drives, modens, softwares, etc.

O que fazer?

No mundo empresarial, banco de dados, processadores de texto, planilhas eletrônicas e geradores gráficos estão entre as ferramentas de produtividade mais necessárias e mais procuradas.

Chamamos a isso de software de produtividade porque são ferramentas que melhoram nossa produção porfissional.

No caso de empresas, dados são importantes quando se quer fazer algum diagnóstico, quando se quer saber alguma coisa, quando se quer tomar uma decisão.

Nestes tempos de concorrência feroz, a correta administração das atividades (produção, recebimento de dinheiro e pagamento de obrigações) é mais do que vital.

Quem tem suas informações sobre custo anotadas em algum lugar consegue fazer o orçamento mais rápido que a concorrência. Quem sabe quanto vai ter que pagar no fim do mês pede mais cedo o desconto de duplicatas ou empréstimo para capital de giro. Quem sabe qual o banco que tem mais duplicatas para cobrar sabe atrás de qual gerente correr na hora do aperto.

Quando se fala em gerenciador de banco de dados é preciso ter em mente três coisas:

- o sistema armazena informações?
- ele modifica as informações quando você decide?
- o sistema recupera essas informações quando você decide?

Numa analogia simples, um Sistema que armazena informações pode ser: uma secretária, um bloco de fichas em branco, um lápis, um arquivo de gavetas, uma tira de etiquetas adesivas.

A parte que modifica as informações fica sendo: uma secretária, um lápis, uma borracha, cola.

E o sistema que recupera informações fica assim: uma secretária paciente, rápida, confiável, que não falte, que não perca a chave do arquivo, resistente, corajosa.

Pois é, se você descobrir uma secretária que seja tudo isso,ela vale seu peso em ouro. Ponha-a no seguro. Vale mesmo.

Mas ela nunca será um computador. Resistente, incansável, rápido, confiável e que usando um gerenciador de banco de dados, como o dBASE II, faz infinitamente mais do que o sistema descrito acima.

Os pequenos e médios empresários precisam que a informática lhes proporcione benefícios imediatos e ao menor custo possível. Assim, é provável que sua empresa não precise de um IBM-PC, pelo menos por enquanto. De qualquer modo, é preciso somar os preços de hardware, software e acessórios, além de avaliar o custo/benefício.

Enfim, o dinheiro a ser empregado vai realmente melhorar os serviços, agilizar as decisões?

Lembre-se que um computador mal adaptado às necessidades ou ao porte de sua empresa não presta serviço nenhum, só atrapalha. E um software sem recursos, ou de recursos escassos não resolve nenhum problema, ao contrário.

À primeira vista pode parecer que se trata apenas de procurar uma boa empresa de informática e comprar os equipamentos necessários. Na maioria dos casos, o interessado encontra uma das muitas lojas que existem nas grandes cidades e pede orientação, sendo que nem sempre consegue o que precisa.

Caso encontre algum programa que pareça interessante, deve pedir para vê-lo em funcionamento, experimentá-lo, ser exigente.

Se precisar de alguma característica e alguém reponder rapidamente que não dá, é bom desconfiar. Geralmente dá, mas a pessoa é que não sabe como fazer e não quer confessar.

Faça o seu papel de consumidor. Afinal de contas, o dinheiro é seu.

É bom lembrar que, por mais que se multipliquem as escolas que ensinam as linguagens BASIC e COBOL, a programação de computadores sempre será uma atividade de especialistas que passam dias, semanas, meses, tentando montar grandes quebra-cabeças.

Trata-se de uma atividade que consome muito tempo, exige dedicação de aprendizado, técnica em constante aperfeiçoamento e um conhecimento profundo de lógica e das linguagens com as quais se trabalha.

Quanto ao Basic, uma linguagem inventada em 1964, está bem longe das necessidades de hoje.

Mas e agora?

Você já optou pelo poderoso hardware MSX. Se na sua configuração tiver pelo menos um disk drive é meio caminho andado. É só adquirir o software.

Vale a pena lembrar a importância da compatibilidade do seu software para MSX com um software para IBM-PC, pois isto facilitará a migração futura, se necessário, dos seus arquivos e da própria cultura do software.

Um exemplo é o dBASE II Plus MSX, que vem a ser o único gerenciador de banco de dados do mercado e que já facilitou e agilizou a vida de muitas empresas e que merece uma atenção especial na hora de se montar um sistema que exige a manipulação de dados.

O MULTIMODEM MSX LIGA VOCÊ AO MUNDO

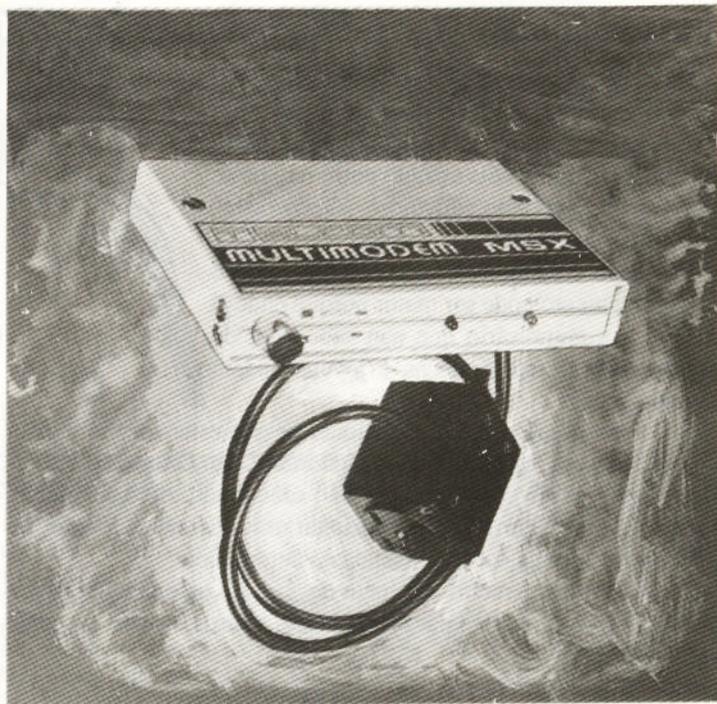
Seu MICROCOMPUTADOR está desinformado, com cara de quem já fez tudo o que podia?

Volte a ativa com o MULTIMODEM DA TELCOM, que lhe deixa na ponta dos dedos o Brasil e o mundo. Você tem acesso a bancos de dados como: CIRANDÃO, BIREME, RENPAC, SAMPA, VIDEOTEXTO, entre outros, através de ligações a grandes sistemas de computadores que também lhe proporcionam, troca de informações, programas e jogos.

Você sai do isolamento, ganha novos amigos e ainda tem à mão o melhor da TELEINFORMÁTICA.

O MULTIMODEM agora tem Discagem Automática (dispensa o uso do aparelho telefônico)

TELCOM INFORMÁTICA LTDA.
Rua Anita Garibaldi, 1.700
CEP 90430 -
PORTO ALEGRE -RS
Telefone: (0512) 41-9871

MSX SOFT INFORMÁTICA (RJ)
Fone: (021) 2340775
Nasa Computadores (SP) Fone (011) 9142266
The Lucky Man (BA) Fone (071) 3595599
Prática Proj. e Assist. Téc. Ltda. (SC) Fone (0482) 220819
e lojas especializadas em MSX

CARTAS

Desejo corresponder-me com pessoas interessadas em trocar dicas e macetes de programação em Basic, Assembler e, principalmente, sobre "C".

ALEXANDRE CARDOSO DULLIUS
RUA LINDOLFO COLLOR 111
CENTRO
93800 - SAPIRANGA - RS

... Aproveito a oportunidade para desejar-lhes muito sucesso com a revista "CPU", de excelente qualidade, feita por pessoas que conhecem o equipamento e que se propõem a escrever.

Como usuário do MSX, gostaria que, se possível, me fossem esclarecidas algumas dúvidas.

Possuo um drive Microsol DRX 360 com interface CDX2

- É possível retirar a fonte, colocando-a separada, pois seu aquecimento é excessivo?
- A ligação da interface é algo assim escabroso, pois a mesma não tem saída na caixa do drive, ficando o cabo dobrado com saída lateral ou reto passando por cima da fonte e correndo o risco de ser cortado pela caixa. Será que liguei errado?
- O comando VERIFY ON/OFF está sempre ativado ou tem que ser reativado a cada operação?
- É possível a instalação de um led para saber quando está ligada?

Meu equipamento é o seguinte: Micro Hot Bit - Drive DRX 360 DD e Interface CDX 2

Sem mais para o momento, agradeço pela atenção dispensada e subscrevo-me,

SÉRGIO LUIZ RUIZ SOLER
RUA DO CORTUME 625 BL C 4
ÁGUA BRANCA
05065 - SÃO PAULO - SP

Enviamos sua correspondência à Microsol que nos enviou a seguinte resposta:

Com relação à carta acima citada, informamos que:

Realmente, este aquecimento não é característico da fonte. Caso seja do seu interesse, V. Sa. poderá enviá-la para manutenção, sem quaisquer ônus. Somente as despesas de frete serão de sua responsabilidade.

A ligação do cabo requer a abertura do gabinete. Essa foi uma medida encontrada para que o preço fosse mais acessível. Entretanto, o drive possui uma depressão entre o gabinete e o acionador que permite a passagem do cabo, sem maiores problemas.

O comando verify encontra-se na página 99, do livro Sistema de Disco para MSX, que segue em anexo, para maiores esclarecimentos.

Cordialmente,

NAGILA LOPES
DEPTO DE COMUNICACÃO
MICROSOL TECNOLOGIA LTDA.
RUA ALMIRANTE RUFINO 779
VILA UNIÃO
60420 - FORTALEZA - CE

... Quanto às várias dúvidas de vossos leitores sobre a transformação do MSX 2, repasso aqui respostas a todas as cartas de uma só vez, mas sem responder a uma de cada vez, já que são muitas cartas e muitas dúvidas. Acredito que assim, poderão poder dar um atendimento maior a todos os vossos leitores de uma só vez.

Preço da transformação: US$ 400
Preço do cartucho MegaRAM: US$ 200
Prazo de entrega: +/- 30 dias
Prazo de garantia: 90 dias

Várias outras dúvidas não me senti capacitado em responder, já que se tratam de consultas que, a meu ver, devem ser feitas diretamente e particularmente ao Ademir, tais como: preços de serviços especiais, se é possível viajar até São Paulo chegando pela manhã e levando o micro de tarde, etc, etc.

PEDRO HENRIQUE GAMA
CAIXA POSTAL 94368
25800 TRÊS RIOS - RJ

Possuo um MSX há algum tempo e, lendo um exemplar da revista CPU de número 3 achei, de muito boa qualidade os assuntos, artigos e programas publicados. Só fiquei um pouco decepcionado de não encontrar a seção de dicas de mil vidas na revista de número 5 e acho que não fui só eu que fiquei assim.

Se possível, gostaria de sugerir que fossem publicados comentários, mapas, dicas, macetes e truques de jogos, para que o usuário consiga concluir o jogo. A meu ver, os leitores não se interessam só por programas complexos. Existem os aficionados por jogos e tenho certeza que não são poucos. Com esse espaço a revista só tem a ganhar.

Gostaria de saber se há condições de receber as revistas de número 1 e 2 pelo correio, para que a minha coleção fique completa.

Deixando um forte abraço, desejo-lhes um Ano Novo com muito sucesso.

SÉRGIO LUIS CESAR
BECO AUGUSTO DOS ANJOS 25
21710 - PADRE MIGUEL - RJ

Realmente você tem razão. No número 5 nos discuidamos um pouco com a parte de jogos, principalmente no que se refere a seção de dicas de mil vidas. Não acreditávamos que esta seção estivesse fazendo tanto sucesso. Estamos incrementando a parte de jogos e já neste número poderá ser notada a diferença.

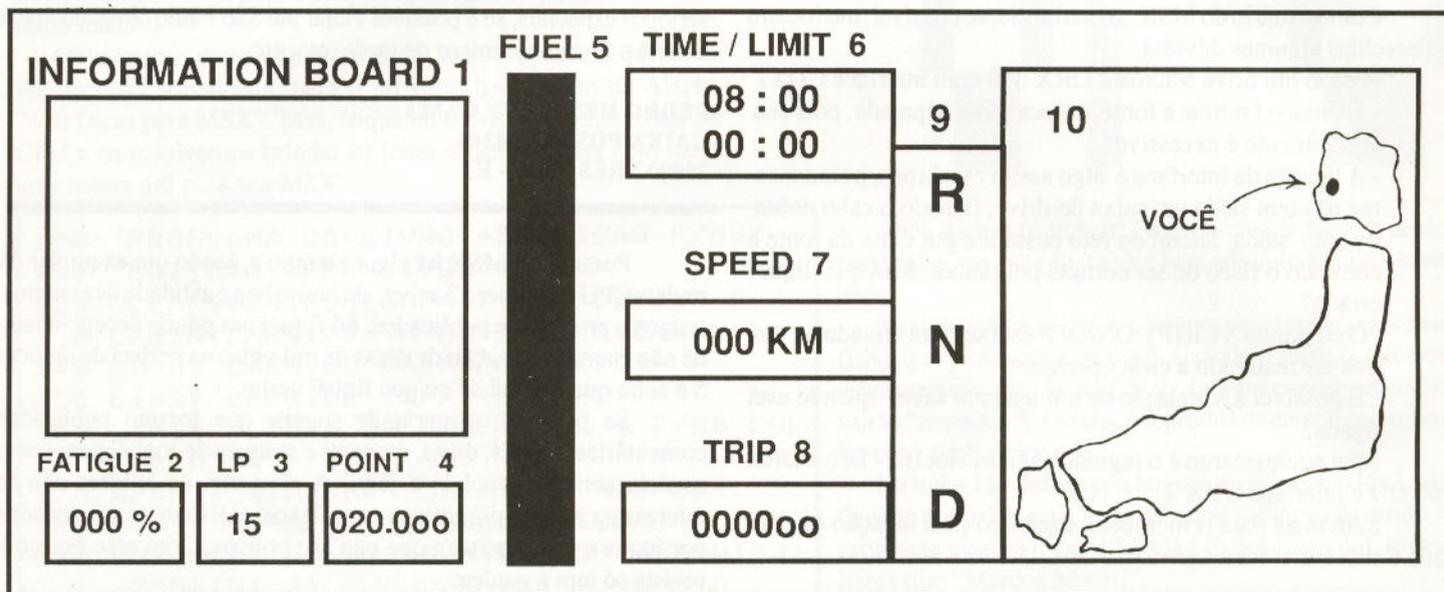
Com relação aos números atrasados, informamos que o número 2 encontra-se esgotado. O número 1 poderá ser solicitado pelo correio, assim como os demais, mediante o envio de cheque nominal, ou vale postal, pagável á agência Copacabana, no valor do custo do número que se encontra nas bancas.

PAY LOAD

PAY LOAD é um jogo sensacional, onde você tem que dirigir um caminhão pelas estradas do Japão . Neste artigo, fornecemos os comandos do jogo e alguns macetes que irão torná-lo um excelente motorista.

LUIZ FERNANDO FIACORDI
PAULISOFT INFORMÁTICA

- Para iniciar, pressione a tecla F2.
- Para movimentar o caminhão pelas ruas, pressione as setas direita e esquerda ou use o joystick.
- Para acelerar, pressione a barra de espaço ou o botão do joystick.
- Para brecar, pressione shift ou graph ou code ou caps ou ainda o botão do joystick.
- Para mudar as marchas, pressione as setas para cima e baixo ou joystick.
- Para gravar ou carregar, presione STOP, ou seja, quando você já estiver cansado de jogar, mas quiser continuar outra hora do mesmo lugar que parou, basta pressionar STOP, escolher a opção SAVE e pressionar a barra de espaços>.
- Agora, se você quiser carregar o jogo do lugar que parou, proceda da mesma forma. Somente pressione STOP após começar o jogo. Escolha a opção LOAD e pressione barra de espaço. Não se esqueça de pressionar a tecla PLAY do gravador.

Obs: Os comandos SAVE e LOAD, mencionados anteriormente, só podem ser utilizados para gravar em fita cassete.

COMANDOS

1 - INFORMATION BOARD (informação de bordo)

Aparece escrito os vários tipos de informações que irão lhe interessar, como, por exemplo, por que motivo e quanto dinheiro perdeu ao ser multado.

2 - FATIGUE (cansaço)

Indica o nível de cansaço do motorista. Quando essa porcentagem ultrapassar 100%, procure um DRIVE-IN. Chegando lá, estacione seu caminhão em cima da faixa escrita TTC. Compre um "MEAL" (refeição) e um "COFFEE" (café). Caso você não proceda desta forma, o motorista ficará tão cansado que não conseguirá dirigir direito, ou seja, não vai obedecer aos seus comandos.

3 - LP (número de infrações que se pode cometer)

Inicialmente, você começa com 15 e, conforme for multado, batendo, o número irá abaixando. Caso chegue a zero, sua licença será cancelada.

4 - POINT (Pontos)

Indica o número de pontos que você tem. No início, você começa com 20 mil pontos. Esses pontos servem para você comprar comida, combustível, peças para o caminhão, etc.

5 - FUEL (Combustível)

Indicador de combustível. Caso a faixa vermelha chegue ao fim, o jogo acabará por falta de combustível. Para encher o tanque, proceda da seguinte forma: procure um posto de gasolina (de preferência nas cidades, pois no campo não há posto), estacione em cima da faixa onde está escrito TTC. O tanque será reabastecido, bastando, para isso, que você não saia de cima da faixa. Enquanto seus pontos estiverem sendo descontados, seu tanque estará sendo abastecido.

6 - TIME/LIMIT (tempo limite)

Existem dois relógios marcando as horas. O relógio superior marca as horas do dia, sendo que, inicialmente, mostrará 8 horas da manhã. Quando chegar às 19 horas, escurece, só amanhecendo novamente às 6 horas da manhã. O relógio inferior indica o tempo que lhe falta para entregar o carregamento. Portanto, não se atrase, pois caso isto aconteça, será descontado do seu pagamento.

7 - SPEED (velocidade)

Indica a velocidade do caminhão.

8 - TRIP (giro)

Indica os quilômetros rodados pelo caminhão.

9 - RND (marchas)

R para ré, N para neutro, e D para a frente.

10 - MAPA DO JAPÃO

O ponto branco indica a sua localização. Caso aceite algum emprego, aparecerá um ponto vermelho, que será o seu destino.

FICHA DE TRABALHO
11 Current Location: Hokkaido
12 Assignmet: Hokkaido Nigata
13 Load: Cow
14 Load Wight: 0047t
15 Delivery Time Limit: 05' 48"
16 Distance: 1300 Km
17 Payment: 13160 pts
18 Accept? YES NO

A ficha de trabalho é obtida no "WORK" de cada cidade, rural ou urbana.

11 - CURRENT LOCATION (local atual)

Indica em que lugar você está. De acordo com o exemplo acima, você está em Hokkaido.

12 - ASSIGNMENT (destino)

Indica a cidade que você está e a seguir o seu destino, ou seja, o lugar para onde deve levar o carregamento.

13 - LOAD (carga)

Indica o que você irá carregar. De acordo com o exemplo, você levará uma carga de vacas.

14 - LOAD WEIGHT (peso da carga)

Indica o peso do carregamento. Cuidado, pois quando o peso exceder, você poderá tomar uma ou várias multas.

15 - DELIVERY TIME LIMIT (tempo limite de entrega)

Indica o tempo que você tem para chegar ao seu destino com a carga.

16 - DISTANCE (distância)

Indica a distância do lugar atual até o destino desejado.

17 - PAYMENT (pagamento)

Indica quanto você ganhará com essa encomenda. É o mais importante de todos. Por isso, não se preocupe com carregamentos de baixo pagamento, pois a viagem poderá sair mais cara do que você pensa.

18 - ACCEPT (aceite)

Se você aceitar o emprego, ponha a mãozinha no YES, caso contrário, ponha no NO. Não se esqueça de, após escolher sua opção, pressionar a barra de espaços.

TIPOS DE MULTAS
- Ultrapassar o sinal vermelho
- Ultrapassar o limite de velocidade
- Ultrapassr o limite de carga
- Advertência
- Dirigir bêbado
- Dirigir na contra-mão

Os dois últimos tipos de multa são os piores, pois sua carteira de motorista será apreendida, dando fim ao jogo. Os policiais que lhe poderão aplicar essas multas usam carros brancos, sendo que também existem radares espalhados pelas ruas das cidades.

HIGHWAY (RODOVIAS)

Servem para que você chegue mais rápido aos lugares. Você poderá pegar uma rodovia de Hokkaido e poderá ir para Aomori, Miyagi, Tochigi, Tokyo, Shizuoka, Aichi, Oosaka, Hiroshima, Fukuoka ou vice-versa. Para tal, você terá que pagar um pedágio, referente à rodovia que escolher.

Nas rodovias será sempre claro, ou seja, nunca haverá noite.

WORK (TRABALHO)

Quando você quiser um emprego, basta encontrar um WORK, estacionar na faixa TTC. Aparecerá uma ficha, com todos os detalhes. Para aceitar uma encomenda, basta escolher o YES e partir para o local determinado. Assim que você chegar no local determinado, leve o carregamento até o WORK.

GS (POSTO DE GASOLINA)

Para encher o tanque, basta posicionar o caminhão em cima da faixa escrita TTC.

Não existem postos de gasolina nas cidades rurais.

DRIVE-IN

Pare em cima da faixa TTC. Aparecerá um MENU de opções, com o seguinte:

19	BEER	1000 pts
20	MEAL	2000 pts
21	COFFE	500 pts
22	EXIT	

19 - BEER (cerveja)

De vez em quando, é sempre bom tomar uma lourinha. Mas tome cuidado para não dirigir bêbado, pois sua licença poderá ser cancelada.

20 - MEAL (refeição)

Como você viaja o dia inteiro e nunca pára, o motorista sente fome. Portanto, se você não alimentá-lo direito, ele ficará tão cansado que não obedecerá a seus comandos.

Obs: O cansaço do motorista é medido pelo FATIGUE.

21 - COFFEE (café)

É muito bom durante a noite, mas também não podemos dispensá-lo durante o dia.

Obs: Tome sempre um café após as refeições. Com isso, o indicador de FATIGUE aumentará lentamente.

22 - Exit (saída)

Volta ao jogo.

PARTS TUNE (OFICINA MECÂNICA)

Pare em cima da faixa TTC. Aparecerá um menu de opções, com o seguinte:

23	GASTANK	10000 pts
24	TURBO	100000 pts
25	RADIO	30000 pts
26	RADER	15000 pts
27	LIGHT	5000 pts
28	EXIT	

23 - GASTANK (tanque reserva de combustível)

É de grande utilidade, pois assim você não se preocupa tanto em ficar enchendo o tanque. Cada GASTANK custa 10000 pts. Dá para comprar até 2 tanques de reserva.

24 - TURBO (turbo)

Também é de grande utilidade, pois aumenta sua velocidade. Cada TURBO custa 100000 pts. Podem ser comprados até 2 turbos.

25 - RADIO (rádio)

Não tem muita utilidade. Para fazer funcionar o rádio, pressione a letra "T". Cada rádio custa 30000 pts, sendo que somente um pode ser comprado.

26 - RADER (radar)

Protege seu caminhão de quase todos os radares. Apenas um pode ser comprado.

27 - LIGHT (luz)

De grande utilidade durante a noite, aumentando sua visão, podendo ser comprado um farol por vez.

28 - EXIT (saída)

Obs: Você está sujeito, a qualquer hora do jogo, a perder alguma dessas peças.

Luiz Fernando Fiacordi, 16, cursa o 3º colegial em São Paulo, com curso de basic e Inglês, trabalhando, atualmente, com entusiasmo, na Paulisoft.

PAY LOAD

ESTRADAS DO JAPÃO

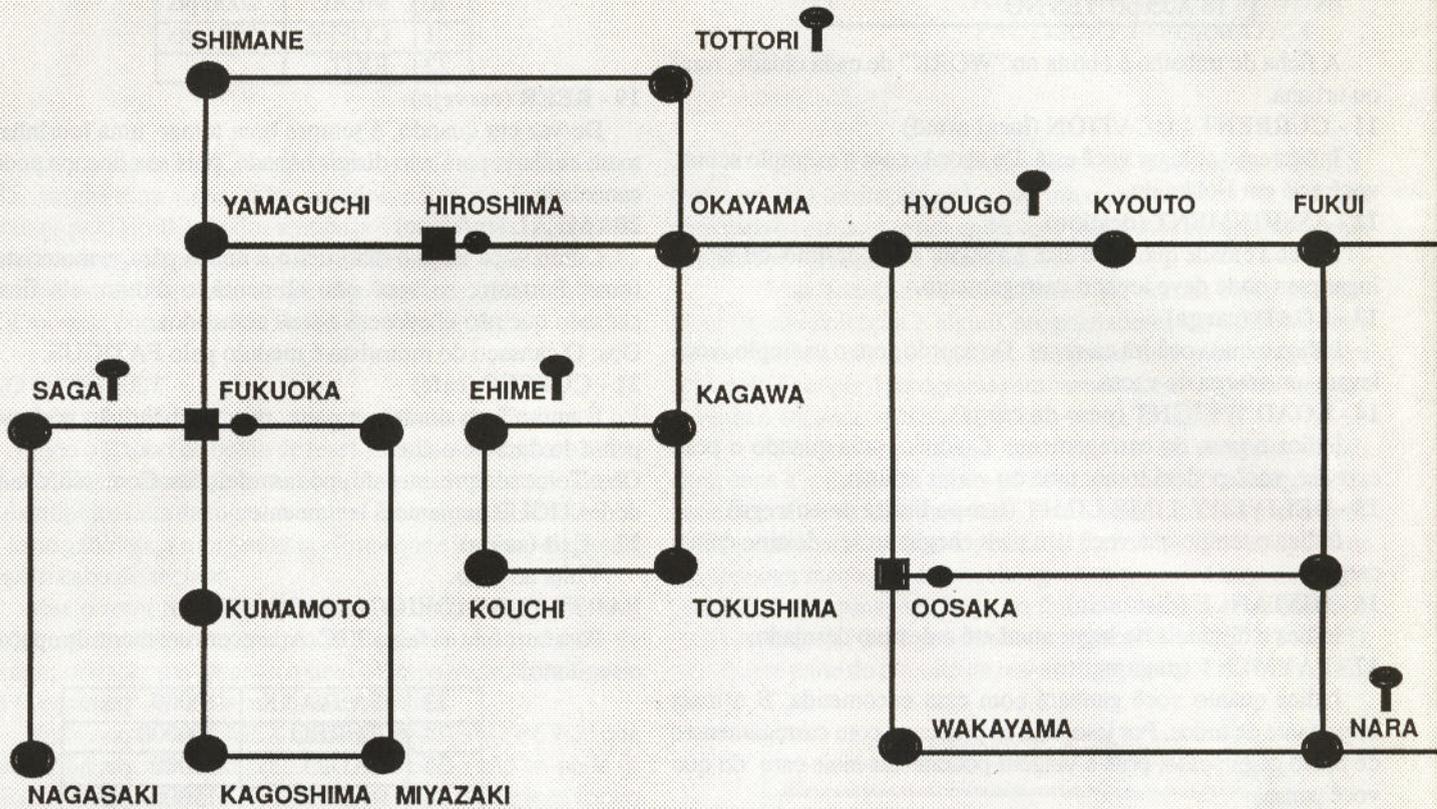
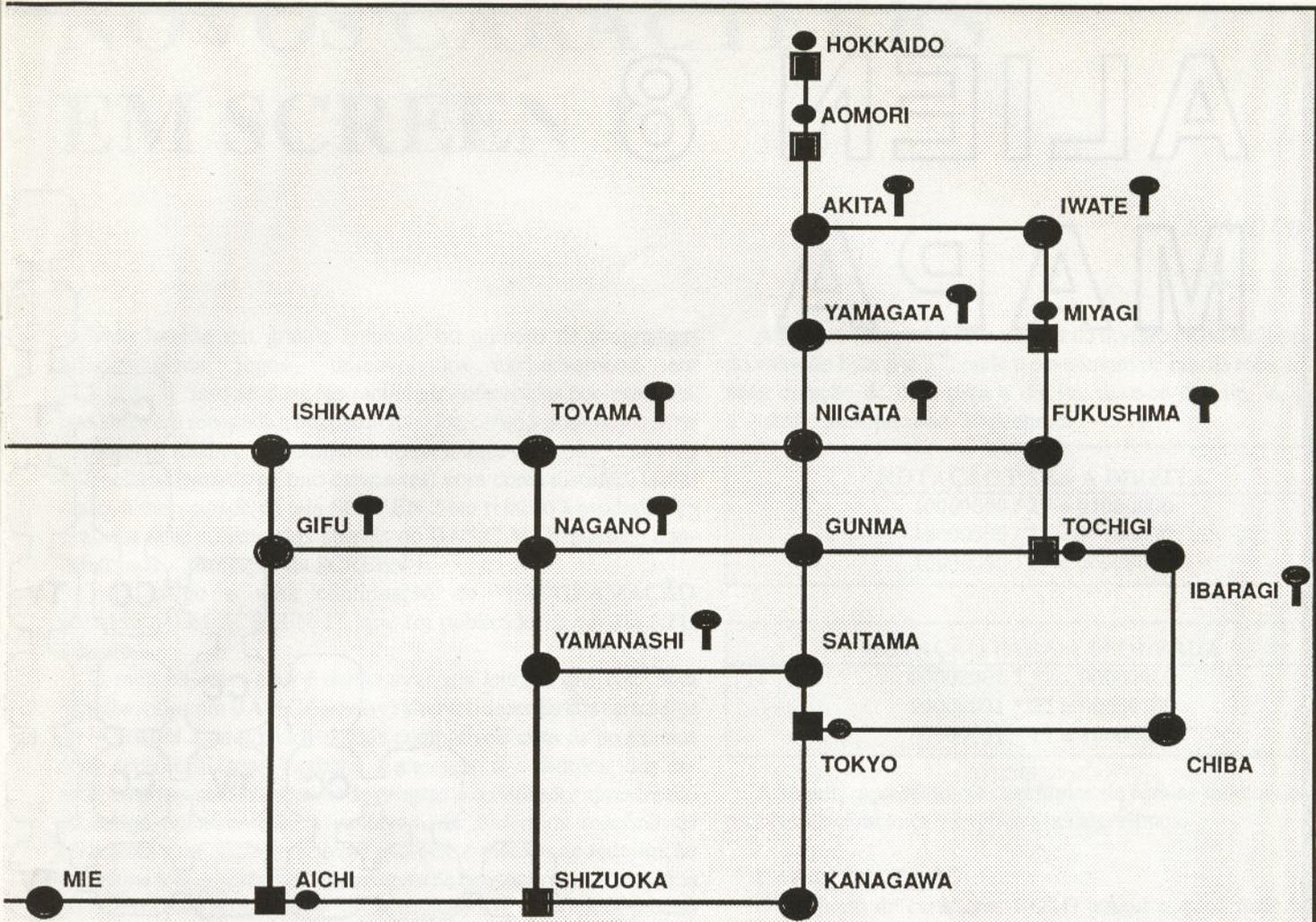



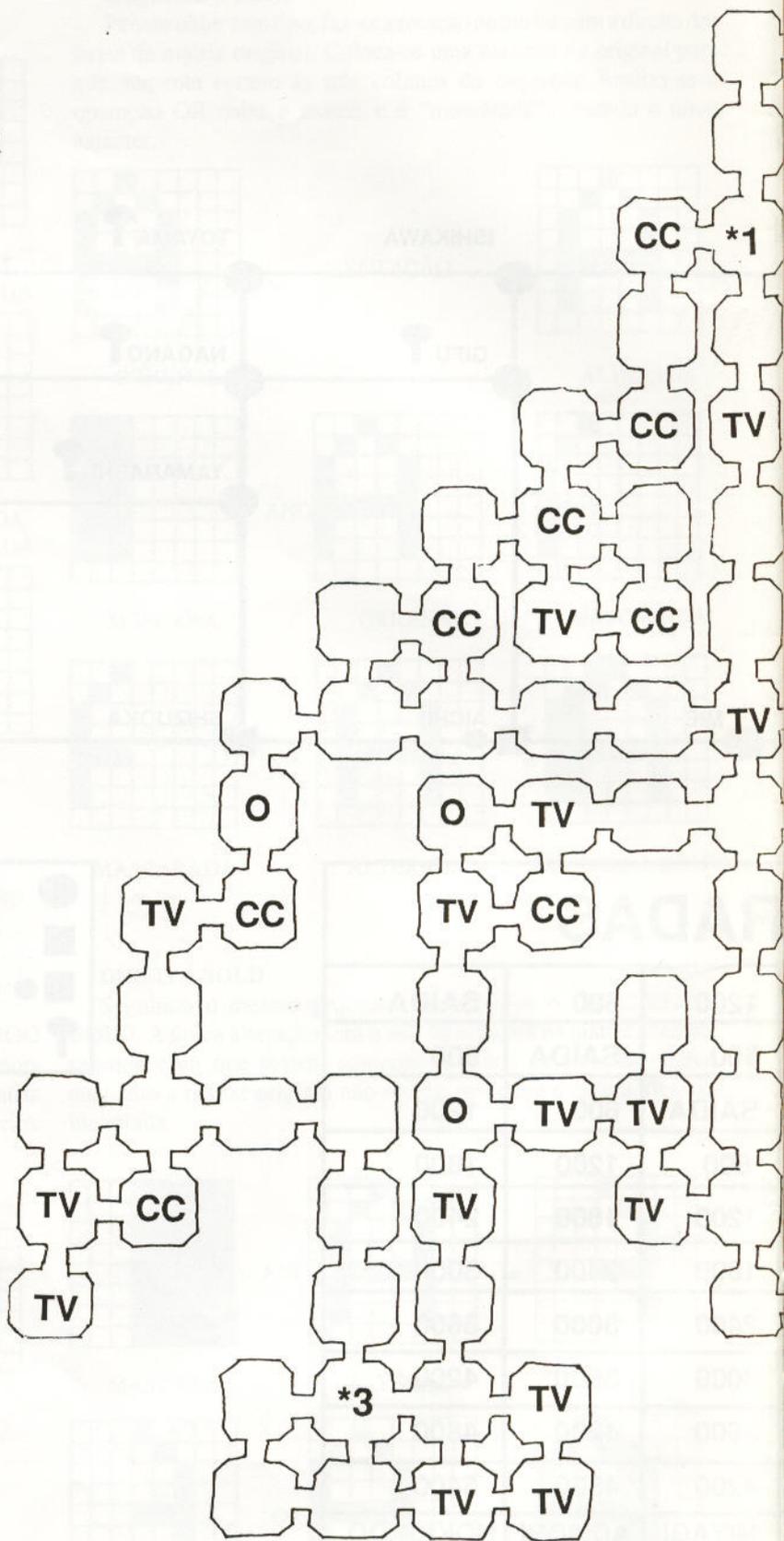
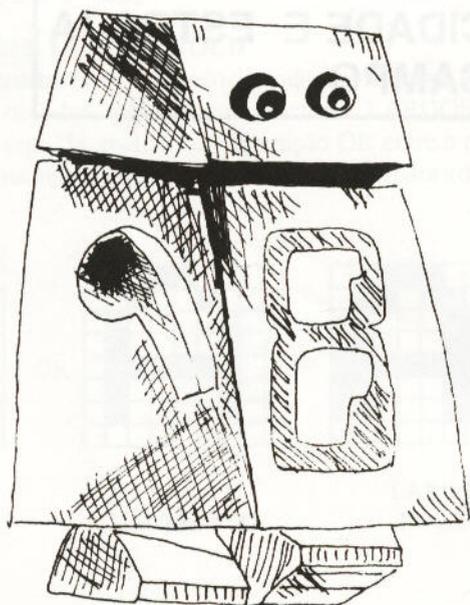
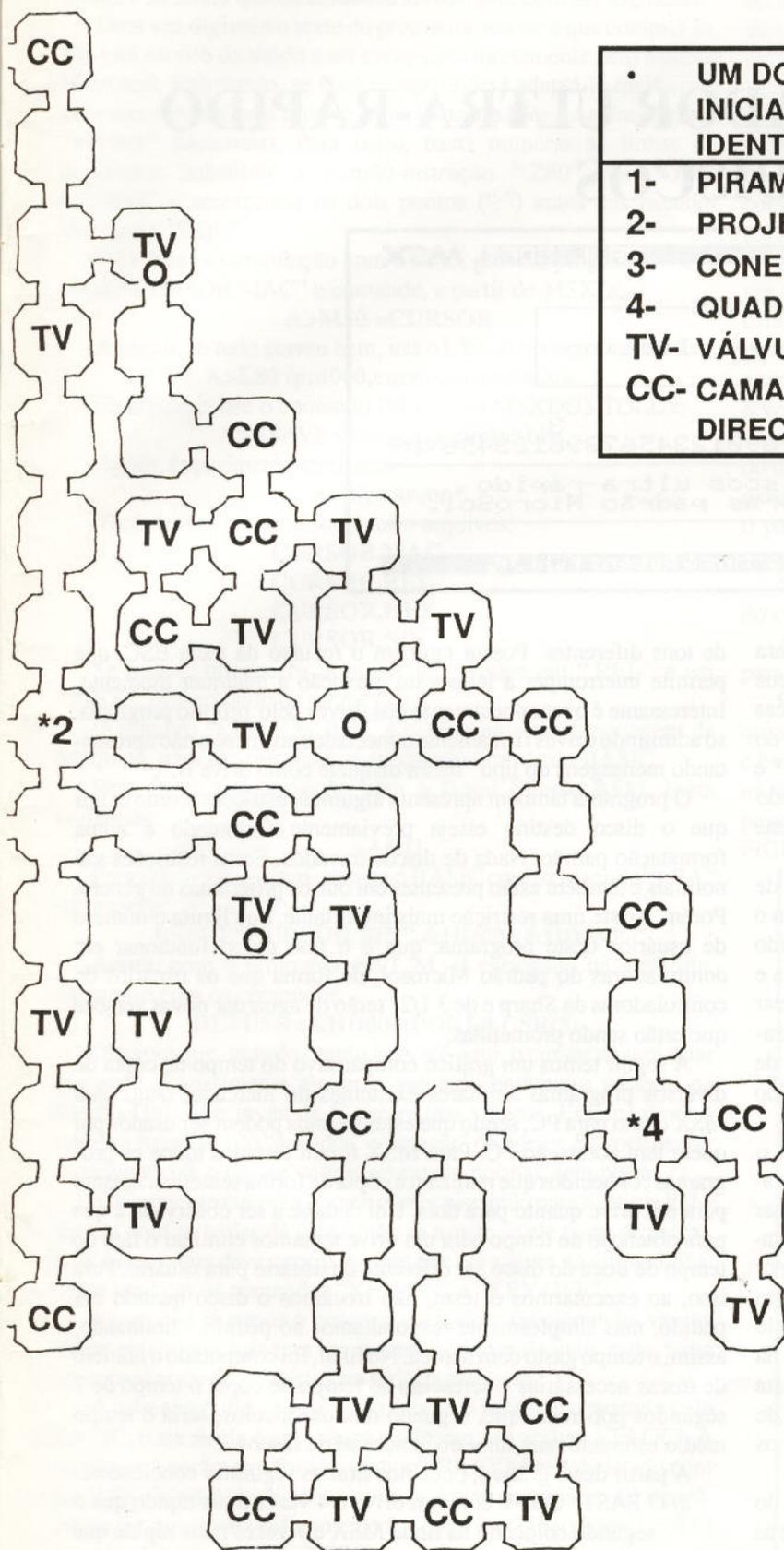
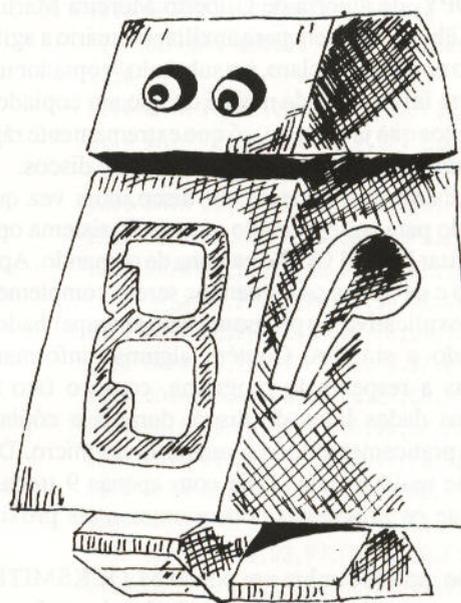
TABELA DE PREÇOS DAS EST

HOKKAIDO	5400	4800	4200	3600	3000	2400	1800
AOMORI	4800	4200	3600	3000	2400	1800	1200
MIYAGI	4200	3600	3000	2400	1800	1200	600
TOCHIGII	3600	3000	2400	1800	1200	600	SAIDA
TOKYO	3000	2400	1800	1200	600	SAÍDA	600
SHIZUOKA	2400	1800	1200	600	SAÍDA	600	1200
AICHI	1800	1200	600	SAÍDA	600	1200	1800
OOSAKA	1200	600	SAIDA	600	1200	1800	2400
HIROSHIMA	600	SAÍDA	600	1200	1800	2400	3000
FUKUOKA	SAÍDA	600	1200	1800	2400	3000	3600
	FUKUOKA	HIROSHIMA	OOSAKA	AICHI	SHIZUOKA	TOKYO	TOCHIGI

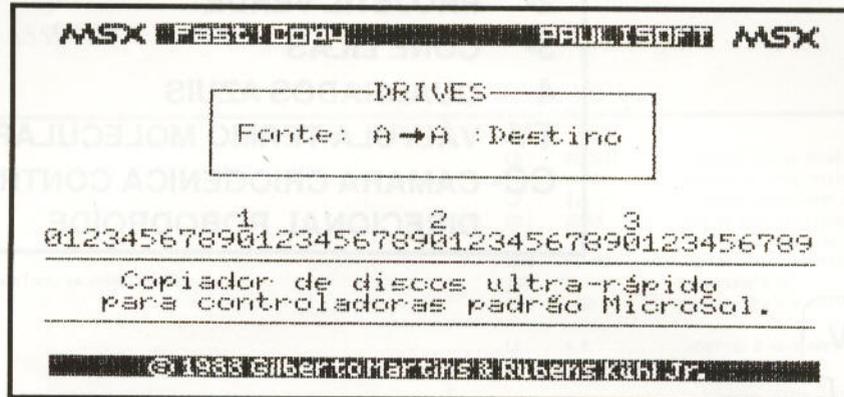
HOKKAIDO
AOMORI
AKITA
IWATE
YAMAGATA
MIYAGI
ISHIKAWA
TOYAMA
NIIGATA
FUKUSHIMA
GIFU
NAGANO
GUNMA
TOCHIGI
IBARAGI
YAMANASHI
SAITAMA
TOKYO
CHIBA
MIE
AICHI
SHIZUOKA
KANAGAWA

RADAS

1200	600	SAÍDA
600	SAÍDA	600
SAIDA	600	1200
600	1200	1800
1200	1800	2400
1800	2400	3000
2400	3000	3600
3000	3600	4200
3600	4200	4800
4200	4800	5400
MIYAGI	AOMORI	HOKKAIDO

ALIEN 8 MAPA

CC
*1
CC
TV
CC
CC
TV
CC
TV
O
O
TV
TV
CC
TV
CC
O
TV
TV
TV
CC
TV
TV
TV
*3
TV
TV
TV

• UM DOS 4 LUGARES ONDE PODE SER INICIADO O JOGO E QUE PODE SER IDENTIFICADO POR:
1- PIRAMIDE AZUL
2- PROJÉTIL VERDE
3- CONE LILÁS
4- QUADRADOS AZUIS
TV- VÁLVULA TERMO MOLECULAR
CC- CAMARA CRIOGENICA CONTROLE DIRECIONAL ROBODROIDE
*2
*4

SOFTWARE

LANÇAMENTO

FAST! COPY COPIADOR ULTRA-RÁPIDO DE DISCOS

O usuário de MSX com um drive sofre alguns problemas para produzir cópias de seus programas, principalmente quando seus discos estão cheios, pois o tempo gasto e o número de trocas necessárias desanimam qualquer um. Sem dúvida, o comando do sistema operacional mais utilizado depois do DIR é o COPY *.* e o usuário sabe bem como sofre com esse comando, até sonhando com a oportunidade de, um dia, comprar um segundo drive, dia este que, para muitos, dificilmente chegará.

A PAULISOFT Informática, dentro de uma nova filosofia de incentivo ao software original e nacional, como demonstrou com o lançamento do EDTRONIC e do SPRITE MAKER, está lançando agora o FAST! COPY, de autoria de Gilberto Moreira Martins e Rubens Henrique Kühl Jr., que vem para auxiliar o usuário a agilizar suas cópias. Como o nome deixa claro, e o subtítulo "copiador ultra-rápido" enfatiza, este utilitário nada mais é do que um copiador de setores como os muitos que já existem, só que extremamente rápido e necessitando um número muito menor de trocas de discos.

O programa é distribuído somente em disco, uma vez que o mesmo foi produzido para ser executado a partir do sistema operacional, bastando digitar FAST1 COPY na linha de comando. Apesar da facilidade de uso e de todos os comandos serem complementados por mensagens explicativas, o programa vem acompanhado por um manual ilustrado e simples. Contém algumas informações técnicas muito boas a respeito do programa, como o fato dele armazenar parte dos dados lidos do disco, durante a cópia, na VRAM, utilizando praticamente toda a memória do micro. Desta forma, ele consegue realizar uma cópia com apenas 9 trocas de discos, enquanto que os programas concorrentes mais próximos necessitam de 16.

Sua apresentação gráfica lembra um pouco o LOCKSMITH do Apple, onde um cursor vai varrendo uma região da tela conforme as trilhas e os lados vão sendo lidos ou gravados, permitindo que o usuário possa monitorar sua cópia. Possui boas indicações de erros de leitura ou de gravação, além de avisar se o disco está com selo de proteção impedindo a gravação, acompanhados sempre de "beeps" de tons diferentes. Possui também o recurso da tecla ESC, que permite interromper a leitura ou gravação a qualquer momento. Interessante é o reconhecimento dos drives pelo próprio programa, só admitindo drives fisicamente conectados ao micro e não apresentando mensagens do tipo "insira disquete como drive B:".

O programa também apresenta algumas restrições, como exigir que o disco destino esteja previamente formatado e numa formatação padrão. Nada de discos travados. Essas restrições são normais e também estão presentes em outros programas do gênero. Porém, existe uma restrição mais importante, que limita o número de usuários deste programa, que é o fato de só funcionar em controladoras do padrão Microsol, de forma que os usuários de controladoras da Sharp e de 3 1/2" terão de aguardar novas versões que estão sendo prometidas.

A seguir temos um gráfico comparativo do tempo de cópia de diversos programas similares existentes no mercado, tanto para MSX como para PC, sendo que estes últimos podem ser usados por quem tem acesso ao PC. Para MSX foram testados todos os programas conhecidos que realizam a cópia de forma semelhante, tanto para um drive quanto para dois. Um detalhe a ser observado é que para obtenção do tempo para um drive tentamos eliminar o fato do tempo de troca do disco ser diferente de usuário para usuário. Para isso, ao executarmos o teste, não trocamos o disco quando era pedido, mas simplesmente respondíamos ao pedido, eliminando, assim, o tempo gasto com a troca. No final, foi computado o número de trocas necessárias e acrescido ao tempo de cópia o tempo de 7 segundos por troca, que, segundo nossos cálculos, seria o tempo médio estimado para uma troca normal de discos.

A partir deste gráfico, podemos tirar as seguintes conclusões:

a) O FAST! COPY com um drive é 4 vezes mais rápido que o segundo colocado na linha MSX e 7 vezes mais rápido que o mesmo programa com dois drives.

b) O número de cópias é quase a metade das exigidas pelo segundo colocado (15 contra 9).

c) O FAST! COPY, rodando num MSX com 64Kb de RAM e 16

Kb de VRAM, com um clock de 3,58 MHZ é cerca de 1,5 vezes mais rápido do que os programas que rodam em PC com 256 Kb de RAM e clock de 4,77 MHZ.

d) Somente o DISKCOPY do PC com 1 drive consegue ser mais rápido que o FASTCOPY, mesmo assim em apenas 7 segundos.

e) Pela diferença entre o tempo gasto para cópia com 1 e com 2 drives, conclui-se que se perde mais tempo na troca de disco num sistema de apenas um drive do que com a cópia propriamente.

f) A única vantagem verificada nos outros programas semelhantes testados, com excessão do PRO KIT ZAPPER, é o fato de funcionarem com outras controladoras.

Muitos têm dito que programas como o FAST!COPY são incentivadores da pirataria, por permitirem a cópia de programas de forma indiscriminada. Porém, deve ficar claro que o FAST!COPY não faz nada a mais do que os outros copiadores de setores não façam, apenas agilizando o trabalho e, como todos os outros, seu objetivo é de auxiliar o usuário a poder manter cópias de segurança de seus discos, protegendo-os dos imprevistos que rodeiam o mundo da informática e que sua utilização para a produção de cópias ilícitas é de responsabilidade de cada usuário.

Produto:	**Fast Copy!**
Autores:	**Gilberto Moreira Martins e Rubens Henrique Küll Jr.**
Distribuição:	**Paulisoft Informática**
Telefone:	**021-228.1313**
Endereço:	**Av. Prestes Maia 241 conj 908 01031 - São Paulo - SP**
Preço:	**1.5 OTN**

Fig. 2

Tabela Comparativa de Tempo de Cópia

Programa	2 Drives	1 Drive	NT	RV2	RV1
PROKIT ZAPPER 1.0	2 drives não funciona	17'22"	16	---	10.0
DISKIT 2.0	7'40"	11'01"	23	13	6.4
COPY *.* B: 0 bytes livres	6'30"	9'58"	15	11	5.8
COPYDISK 1.2 Nemesis	2 drives não permitido	9'37"	16	---	5.5
COPIED Solxdos 1.2	4'12"	7'33"	25	7.2	4.4
DISKCOPY Dos Tools	4'09"	7'04"	22	7.1	4.1
PCTOOLS A3.23 *	1'18"	2'33"	11	2.2	1.5
LOCKSMITH-PC 1.3 *	56"	1'47"	7	1.6	1.0
DISKCOPY PC *	1'03"	1'31"	3	1.8	0.9
FAST COPY	35"	1'44"	9	1.0	1.0

*** PX XT 4.77 MhZ 256 kb PC-DOS 2.1 NT=Número de trocas. RV=Relação de velocidade. Tempo médio de 7 s. para cada troca.**

SISTEMA OPERACIONAL BKP DISCO

Detalhe do "Copia arquivo"

A grande maioria dos usuários da linha MSX que trabalhou ou trabalha com fita cassete, já utilizou o programa BKP, que vem a ser uma ferramenta que permite ao usuário efetuar cópias de segurança dos programas gravados em fita, mesmo que originalmente eles tenham sido gravados com proteção.

Na época do lançamento do BKP para fita, não existia no Brasil uma lei que regulamentasse a produção e venda de software para computador e o BKP foi vendido livremente, tendo sido, com certeza, um dos campeões de venda algum tempo atrás.

Mesmo sem ter tido o seu trabalho recompensado, o autor do BKP lançou a versão deste programa para disco e deixou a cargo da Paulisoft, softhouse de São Paulo que comercializa, entre outros títulos, o Edtronic e o MSX Turbo a venda do Sistema Operacional BKP de Disco, que também poderá ser encontrado em outras SOFTHOUSES, a exemplo do que acontece com todos os produtos da PAULISOFT.

Mas o que vem a ser o Sistema Operacional BKP Disco?

O BKP Disco é um sofisticado sistema de cópia e edição, formado de 12 comandos que permitem ao usuário trabalhar com a unidade disco de uma forma mais fácil e segura.

Os 12 comandos do BKP Disco permitem produzir cópias de segurança de discos, mesmo que estes possuam proteção contra cópia, manter um arquivo de programas, ou seja, uma lista geral de todos os programas ou arquivos que o usuário possua, facilitando, desta forma, a sua localização, cópia ou edição, ordenar o diretório, restaurar um arquivo que tenha sido apagado acidentalemnte ou que tenha tido algum problema na gravação, efetuar pequenas alterações em arquivos, possuindo, portanto, a função Zapper e diversas outras.

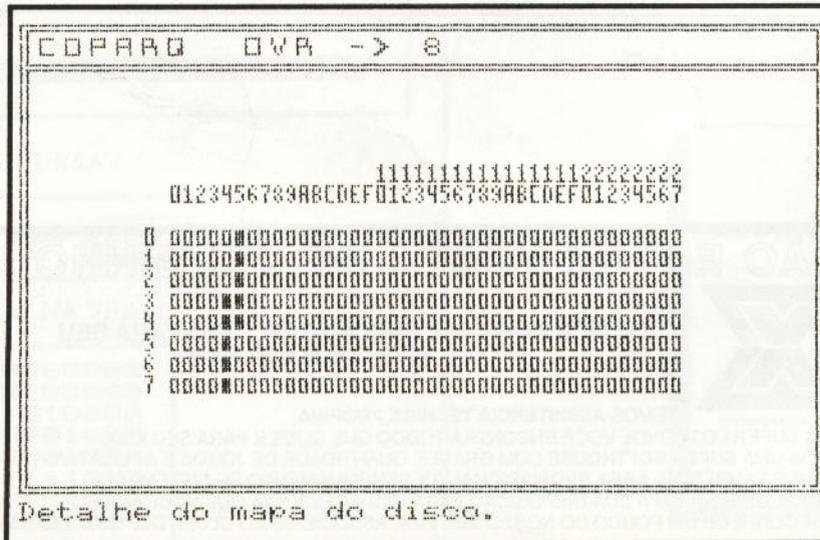
Detalhe do mapa do disco.

O programa é composto de um programa principal (BKP.COM), que tem a função de carregar e executar os demais módulos que compõem o menu principal.

O BKP disco possui três tabelas de caracteres, sendo que o usuário deverá selecionar uma delas, dependendo do computador e versão utilizados (Expert 1.1 ou 1.0 e Hotbit), permitindo também a operação com cartão de 80 colunas.

O manual que acompanha o soft é bastante detalhado, não deixando o usuário com dúvidas sobre a operação do programa, tendo tabela de mensagens de erro e um glossário de termos.

Conclusão:

O Sistema Operacional BKP de Disco apresenta inúmeras vantagens, tanto para os usuários de nível mais avançado como para aqueles que não irão utilizar todas as suas facilidades, como a de alterar arquivos gravados no disco. Uma vez que o usuário comece a trabalhar com o programa, absorvendo todos os seus comandos e maneira de trabalhar, ficará difícil operar com outros sistemas mais lentos e, de certa forma, mais complicados.

1ØØ DICAS PARA MSX

TÉCNICAS E TRUQUES DE PROGRAMAÇÃO

Editora Aleph

linguagem de máquina
ASSEMBLY Z-80
MSX
Rossini - Figueredo

Nossos livros podem ser encontrados em livrarias e lojas de computação. Se o seu livreiro ou fornecedor habitual não os tiver disponíveis, entre em contato conosco pelo telefone (011) 843-3202.

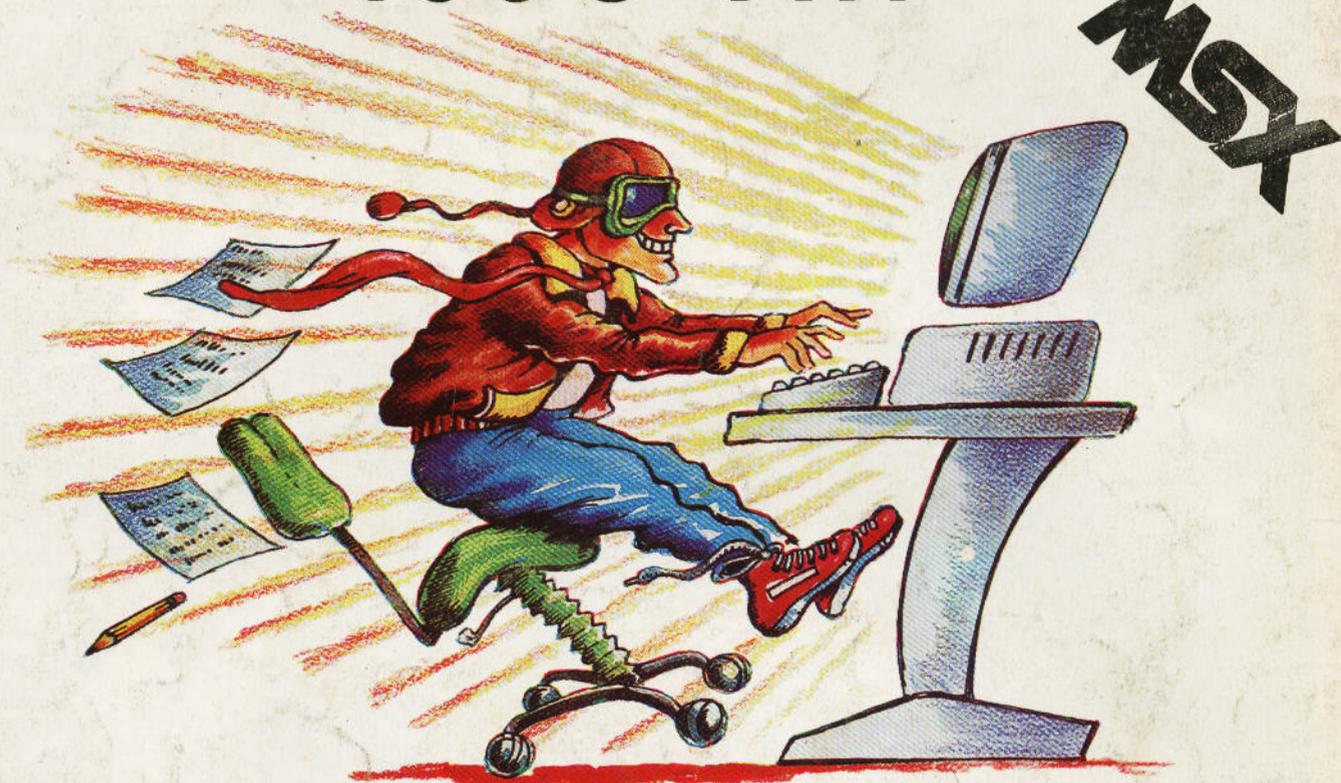
Se você não está recebendo seu boletim gratuitamente pelo correio, ou tem algum amigo que gostaria de recebê-lo, não deixe de enviar o cupom abaixo à EDITORA ALEPH - C.P. 20707 - CEP: 01498 - SÃO PAULO-SP.

NOME: ....

END.: ....

CEP: .... CIDADE: .... UF: ......

TEL: (.....) .... MICRO(S) QUE POSSUI: ....

A NOVA NEWSOFT É ISSO AÍ:

A PARTIR DE AGORA O SEU MICRO FICARÁ MUITO MAIS "INTELIGENTE".

A MELHOR E MAIS BEM APARELHADA SOFTHOUSE DO RIO DE JANEIRO, AGORA EM NOVAS INSTALAÇÕES, COLOCA A SUA DISPOSIÇÃO A MAIS COMPLETA LINHA DE PRODUTOS PARA O SEU MSX!

VOCÊ SÓ TEM A GANHAR !

° SÃO MAIS DE 2.000 PROGRAMAS A SUA DISPOSIÇÃO, COM GRAVAÇÃO PROFISSIONAL GARANTIDA.
° DRIVES, INTERFACES, EXPANSORES, PLACAS, MODEMS, GABINETES, ETC.
° LIVROS E REVISTAS ESPECIALIZADOS.

E MAIS !

AGORA VOCÊ CONTARÁ TAMBÉM COM A EXCLUSIVIDADE DO ... NEWSOFT INFOCARD, O CARTÃO QUE ESTÁ REVOLUCIONANDO O MERCADO DE INFORMÁTICA !

NÃO PERCA TEMPO ! ★ **INFORMÁTICA** ★

Visite **SHOW ROOM** na Av. NILO PEÇANHA nº 50 sala 906, **Ed.Rodolfo de Paoli, CEP** 20020, bem no centro do RIO (Largo da Carioca)